Security and Communication Networks

Anomaly Detection Technologies for Securing the Emerging Resource-constrained Networking Scenarios

Lead Guest Editor: Kai Wang
Guest Editors: Yulei Wu, Xiaofan Liu, Yuanlong Cao, and Hui Xia

Anomaly Detection Technologies for Securing the Emerging Resource-constrained Networking Scenarios

Security and Communication Networks

Anomaly Detection Technologies for Securing the Emerging Resource-constrained Networking Scenarios

Lead Guest Editor: Kai Wang
Guest Editors: Yulei Wu, Xiaofan Liu, Yuanlong Cao, and Hui Xia

Chief Editor

Associate Editors

Academic Editors

Contents

Hindawi
Security and Communication Networks
Volume 2023, Article ID 8903980, 10 pages
https://doi.org/10.1155/2023/8903980

Research Article

An Anomaly Detection Approach Based on Integrated LSTM for IoT Big Data

Chao Li,[1] **Yuhan Fu**,[1,2] **Rui Zhang**,[3] **Hai Liang**,[2] **Chonghua Wang**,[4] **and Junjian Li**[1]

[1]*Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510700, China*
[2]*Guangxi Key Laboratory of Cryptography and Information Security, Guilin 541004, China*
[3]*COMAC Shanghai Aircraft Manufacturing Co., Ltd, Beijing 100005, China*
[4]*China Industrial Control Systems Cyber Emergency Response Team, Beijing 100040, China*

Correspondence should be addressed to Chonghua Wang; chonghuaw@live.com

Received 17 October 2022; Revised 8 December 2022; Accepted 25 April 2023; Published 30 May 2023

Academic Editor: Yulei Wu

Due to the expanding scope of Industry 4.0, the Internet of Things has become an important element of the information age. Cyber security relies heavily on intrusion detection systems for Internet of Things (IoT) devices. In the face of complex network data and diverse intrusion methods, today's network security environment requires more suitable machine learning methods to meet its security needs, and the current machine learning methods are hardly competent. In part because of network attacks by intruders using cutting-edge techniques and the constrained environment of IoT devices themselves, the most widely used algorithms in recent years include CNN and LSTM, with the former being particularly good at extracting features from the original data space and the latter concentrating more on temporal features of the data. We aim to address the issue of merging spatial and temporal variables in intrusion detection models by introducing a fusion model CNN and C-LSTM in this paper. Fusion features enhanced parallelism in the training process and better results without a very deep network, giving the model a shorter training time, fast convergence, and computational speed for emerging resource-limited network entities. This model is more suitable for anomaly detection tasks in the resource-constrained and time-sensitive big data environment of the Internet of Things. KDDCup-99, a publicly available IBD dataset, was applied in our experiments to demonstrate the model's validity. In comparison to existing deep learning implementations, our proposed multiclass classification model delivers higher accuracy, precision, and recall.

1. Introduction

Internet of Things (IoT) is a new network system consisting of a cloud data center and subnodes under it that integrates computing, controlling, and communication technologies. In the era of industry 4.0, wireless network technology and diverse smart devices are increasingly applied to the Industrial Internet of Things (IIoT), and more and more industrial applications are interactively connected through the intelligence and real time of signal processing. Through a large number of distributed IoT devices, ubiquitous sensors are deployed throughout real scenarios. They detect environmental data through various types of sensors and transmit them to processing centers through various types of IoT transmission protocols. The processing center uses cloud computing and big data technologies to extract valuable information from this data and upgrade services. IoT has been frequently employed in various fields such as healthcare, smart home, and intelligent transportation. By 2024, IoT is anticipated to reach 83 billion devices [1]. The diverse category of IoT devices will set off the IoT architecture for innovation.

In addition, cybercrime is growing dramatically in size, complexity, and cost [1] due to the increasing spread of IoT devices with distributed and large numbers in individual homes, national grids, smart cars, and industrial assembly lines, and the complexity of IoT defending systems [2]. Table 1 lists several typical cyber-attacks. The rise of various old and new types of cyber-attacks signifies that the use of resume firewalls and signature certificate-based defending is

Table 1: Several common types of network attacks.

Attacks	Quantity
DoS and DDoS	DoS attack is designed to overload system resources to the point where they can no longer respond to legitimate service requests. And DDoS is initiated by controlling a large number of hosts infected with malware
MITM	As an "indirect" intrusion attack, a man-in-the-middle (MITM) type of network attack allows an attacker to eavesdrop and steal communications from two computers without directly affecting the network
DNS spoofing	Spoofing through the domain name system (DNS) is also a form of man-in-the-middle attack, where a hacker can change the DNS records returned to the querier to a response record of the attacker's choosing
URL resolution	Through URL interpretation, an attacker can change and forge certain URL addresses and access to personal or company private data
Zero-day attacks	Zero-day attacks are computer vulnerabilities that have not been discovered by security vendors but may be in the hands of hacker groups, and once they are discovered, 0 day vulnerability attacks can spread rapidly

in great demand, and instead, a proactive approach must be taken to discover threats. Intrusion detection systems have become a crucial tool for identifying and defending against network attacks in the form of malicious network traffic as security threats continue to spread across the Internet. By extracting features for analysis of network traffic and alerting once unsafe traffic is detected, intrusion detection systems enable network monitoring [3].

Anomaly detection is recognized as one of the key tools for dynamic network security threat detection [4]. There are numerous methods available for network anomaly detection. To improve the performance of anomaly detection systems when processed by intrusion detection systems, artificial intelligence techniques are applied to various types of active-defending systems. However, reliable anomaly detection of massive and complex multidimensional data in industrial IoT is still a tricky task. In recent years, deep learning has excelled in various classification tasks, but a large amount of classification variables in the network stream complicates the anomaly detection process using gradient descent methods. Even though there are numerous methods for anomaly detection, most people do not try to use CNN (convolutional neural network) for anomaly detection, compared to machine learning. As research has intensified in recent years, deep learning has increasingly emerged in the field of complex high-level data processing, such as image and signal processing. Deep learning interprets the internal rules and data expressions of data such as word, image, and sound by extracting the internal features of sample data during the learning process. The ultimate goal is that deep learning models have the ability to analyze and learn from input data and eventually recognize data such as characters, images, and sounds. Among them, two models are widely used: recurrent neural networks (RNN) that mainly extract time-step features for problems of NLP and voice recognition, and CNN with powerful spatial feature extraction for image classification and regression. Zeiler [5] visually understands the functions of the intermediate feature layer and the operation of the classifier through the large convolution neural network model, indicating that CNN is very sensitive to the local structure of data. CNNs reflect the spatial properties of data by extracting spatial cues such as color, level, and edges in images using convolutional perceptual fields and shared weight coefficients while RNN uses gate units to efficiently simulate serialized data to reflect the temporal properties of the data. The LSTM method is mentioned in all model surveys for univariate and multivariate time series data mentioned by Lindemann et al. [6]. To achieve high performance, Kim and Cho [7] constructed a C-LSTM network. They first used preprocessing to initially construct temporal correlations of the dataset, then used CNN to extract these features, and finally used LSTM to extract spatial and temporal features. To ensure that the features extracted by CNN are potentially correlated and more effective than the temporal features extracted by LSTM, Preciado-Grijalva and Iza-Teran [8] used two sliding windows to generate time-dependent subsequences based on C-LSTM. Meanwhile, Yin employed a modified LSTM-based self-encoder to extract more anomalous features from the input sequence. Although CNN and LSTM are both part of their network, the input that LSTM accepts only comes from CNN extraction, and the spatialized extraction of CNN disrupts the temporal aspects of the original data at the potential level, which impacts LSTM's learning effect to some extent.

We propose an improved network structure based on the study of the interaction between CNN and LSTM direct serial methods for extracting data features. The network consists of a CNN and a C-LSTM using temporal convolution, both of which receive input from the original dataset, and the CNN and C-LSTM will focus on extracting spatio-temporal features in the intrusion data, respectively, with the modified C-LSTM using one-dimensional temporal convolution and the LSTM focusing more on purely temporal features of the intrusion data while ignoring some spatial features; the CNN will learn more to reconstruct the spatial features of the intrusion data and do parallel and fusion between the two instead of serial, which can improve the performance of the detection model. In the absence of deeper network depth, anomaly detection achieves higher scores in various metrics. The significant contributions of this paper are as follows:

(1) A network model based on our C-LSTM and CNN fusion is proposed to detect intrusion data using shallow, small-scale deep learning models. The model utilizes the ideas of C-LSTM and exploits its advantages. To retrieve temporal aspects of the intrusion data on a partially parallel model, a one-dimensional convolution is employed in place of the two-dimensional convolution in the original C-LSTM.

(2) Fusing our C-LSTM and CNN to obtain a balance on temporal and spatial features. Compared to the original C-LSTM which presents the final prediction results and scores by a single model, our fusion model is trained and predicted independently by two models. The learning method of fusion learning determines that it does not extract features by CNN and then learn features by LSTM but fuses the features learned by C-LSTM and CNN separately, which extends the dimensionality of the features. In this way, even if the features extracted by the CNN are insufficient, the C-LSTM can be supplemented and extended. In terms of the evaluation of several classification metrics, this model outperforms the C-LSTM model.

(3) In light of the model fusion learning method, its two-part model does not require a deep model depth to learn every feature of the entire dataset and is faster in its training and convergence than other methods. The two parts of our network, C-LSTM and CNN, only need to learn the sensitive part of the data features and combine them for prediction, rather than learning all the features separately. Such a mechanism facilitates its performance on resource-constrained devices.

2. Related Work

This section summarizes machine learning methods for anomaly detection based on a review of researchers' research on anomaly detection or intrusion detection in recent years. As shown in Table 2, in the field of anomaly detection, many researchers classify a segment of the data as normal or anomalous by extracting contextually relevant information from the data. However, in general, this discriminative approach also requires modeling of the data system, so detection methods are divided into three categories: modeling based on data statistics, modeling based on temporal features, and modeling based on spatial features. Either the error in the context of the data predicted by the time series is used as the core of detection [9], or the original data is reconstructed without a priori knowledge, and normal and abnormal values are defined using thresholds [10]. The core of all these detection methods is to extract the necessary features from the original data space that affect its determination as normal or abnormal and thus perform a credible classification.

2.1. Issues on Intrusion Detection System. Researchers have already done in-depth research in the area of intrusion detection for cybersecurity of IoT, cloud data centers, and blockchain systems. They have investigated the general anomaly detection problem by discussing anomaly detection in time-series data in short chapters.

Table 2: Several abnormal traffic detection methods.

Authors	Model	Dataset	Year	Score
G. Bae	CNN	KDD99	2019	Acc = 97.34
A. Diro	LSTM	AWID	2018	Acc = 98.22
Q. Tian	Svm	UNSW-NB15	2019	Acc = 97.00
Y. N. Kunang	DNN	NSL-KDD	2021	Acc = 83.33
In-young	C-LSTM	Webscope	2018	Acc = 99.62
Chunyong Yin	C-LSTM-AE	Webscope	2021	Acc = 98.6

Hawkins [11] and Abraham and Chuang [12], as early developers in this field, have conducted in-depth research on the network security of wireless network, Internet of Things, blockchain system, and other network systems, especially the network security intrusion problem. However, many researches on anomaly detection have found this kind of problem and usually discuss the intrusion problem highlighted by abnormal data in several sections. Markou and Singh [13, 14] have published research showing that among the intrusion detection methods up to 2003, the intrusion detection system based on feature extraction has been widely used these days. Stephen and Arockiam [15] designed a protocol suitable for resource-constrained nodes but with lossy routing and explored an integrated approach to detect Sybil attacks on the IoT.

2.2. Unsupervised Method in Anomaly Detection. Münz et al. used the unsupervised learning method of K-means [16]. They discussed and derived the prime number of clusters by statistical methods. By calculating the prime number and the spatial distance of each flow data, the distance data are used as the standard for distinguishing abnormal and normal data. Zhang and Zulkernine adopted the random forest method based on unsupervised learning [17], calculated the closeness in each case, and designed a mathematical standard based on statistics to distinguish normal and abnormal data.

2.3. Machine Learning in Anomaly Detection. Kaur et al. [18] used CNN models to detect attacks in data streams. They trained and validated their model with the cicids2017 and cicids22018 datasets. Although their approach covers a wide range of intrusion data types, their performance metrics fall short of practicality. To detect intrusions in a massive data environment, Hassan et al. [19] designed an integrated deep learning model using CNN and *wdlstm* (long-term memory with decreasing weights). CNN was used to find the best features, and *wdlstm* method was used to prevent overfitting in neural networks [20]. The Bayesian neural network is studied in, and the LSTM based self-encoder is used to replace some previous data extraction and analysis structures, and then, the MLP is used to perform the final prediction step. Chen and Lin constructed time-step features of the original data using a sliding window preprocessing algorithm. Then LSTM models were used to extract information on the preprocessed high-dimensional data [21]. The LSTM model studied by Malhotra et al. [9] was based on sensor data and normal signals. They used the trained LSTM model to predict the succeeding signal as a criterion for judging, and then, the actual input and the criterion were calculated to derive the error distribution for anomaly detection.

These methods use models that model serialized data to learn temporal features and thus have the ability to predict predictive classification. Just like RNN, after training RNN, RNN will have the ability to predict future data. In this way, the output of the model can be used as a criterion to compare with the actual data. The result of the comparison will be determined by a set threshold value to determine whether it is normal or abnormal. This approach is based on periodic traffic data and performs better on its dataset. However, if periodicity is not predominantly represented on the data set, the predicted results will not be accurate for the actual data.

As mentioned previously, many attempts have been made to detect intrusion data using various methods. However, few attempts have been made to focus on spatio-temporal features of the data to conduct research. Most studies, in order to improve the performance of a single dataset, usually model only the key features in the datasets that have a high degree of impact, while few studies have talked about taking both temporal and spatial features into account. In contrast, spatio-temporal information is crucial for data analysis and reconstruction because it integrates the spatio-temporal features of the original data. In order to utilize both temporal and spatial information in complex traffic data for anomaly identification, a suitable learning method is therefore required.

3. Modeling of C^2-LSTM

Using models that make compromise judgments on temporal and spatial features, C^2-LSTM is a modified C^2-LSTM model designed to evade attacks to achieve deception of resource-constrained models by intrusion data. In the C^2-LSTM, the CNN and the improved C^2-LSTM learn the spatial and temporal dimensions of the intrusion traffic, respectively.

3.1. Problem Definition. Set $D = \{X^1, X^2, \ldots, X^k\}$ is a input set which represents k kinds of labeled network traffic data in anomaly detection. k includes q kinds of anomaly samples and p kinds of normal samples.To be specific, p is much bigger than q, and $p + q = k$. $Y = \{y^1, y^2, \ldots, y^k\}$ is a set of label results for input D. The work in [22–24] tries to find outliers in the data and edit to identify them, while the work in [25, 26] is for the classification and labeling of the target sequence, whether it is abnormal or normal. The former is a regression method that regresses the input data into an exact value. The latter is a clustering or classification problem to classify it into one of the predefined categories. The issue studied in our paper is a classification problem, i.e., classifying the input data into a corresponding type y^1.

3.2. Our C-LSTM. CNN and LSTM are the two components of C-LSTM. Similar to C-LSTM, he uses self-encoders based on LSTM and CNN to extract fused spatio-temporal features from the data. Figure 1 shows its model structure.

The C-LSTM uses preprocessed data as input. The convolutional layer uses convolutional kernels for learning and feature extraction, and the parameters of each layer are optimized by a back-propagation algorithm. Convolutional operations can extract various features from the data space level. The first few layers of convolution may only extract some low-level features. For images, these are picture corners, single lines, edges of objects, etc. that are not sensitive to the impact of the results, while higher level features that affect the model performance will be extracted north in the deeper layers of the network. The pooling layer reduces the computational effort by partitioning and sampling the data, down sampling a large matrix into a smaller one, and can prevent overfitting at the same time. The feature data are transported to the LSTM. First, the CNN is composed of a convolutional layer and a pooling layer for automatically extracting a sequence of high-level spatial features of the network traffic. We use a one-dimensional convolution operation to extract the temporal features of the input data directly by temporal convolution instead of the normal two-dimensional convolution of C-LSTM. After the convolution, an activation function is used to perform the transformation of the non-nonlinear function. As a result, the model is able to capture features of more dimensions.

Suppose it is an input vector of intrusion data and n is the dimensionality of its features. Equation (1) yields the output value from the i-th convolutional layer.

$$y_i = \sigma\left(b_i + W_i \bullet x\right), \tag{1}$$

where b is the bias of the feature mapping, W is the weight of the kernel, and σ is an activation function.

We use circular units running from left to right to enable the LSTM layer to understand the temporal properties of the traffic data extracted from the upper CNN layer. This makes the model in this layer to have a stronger understanding of the feature transformation relationships on the time scale. His input is the output of the pooling layer of the upper layer, which is gated to control the discarding or adding of information for forgetting or remembering. Gating is an information selective pass-through structure based on a multiplicative mechanism, consisting of a sigmoid function and a dot product operation that updates the cell state of each gate according to its activation. Sigmoid functions have output values in the interval [0, 1], with 0 representing complete discard and 1 representing complete pass-through. Cell management through these gates handles the upper layer of input to input, output and forgetting gate operations. The hidden value of the LSTM cell, h_t, is updated once per step t.

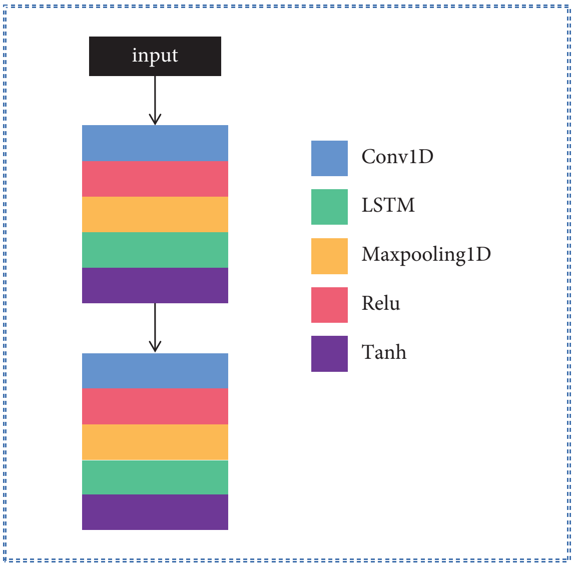


Figure 1: Model structure of our C-LSTM which consists of one-dimensional convolution, LSTM, and one-dimensional maxpooling.

$$f_t = \sigma_g\left(W_f x_t + U_f c_{t-1} + b_f\right), \tag{2}$$

$$i_t = \sigma_g\left(W_i x_t + U_i c_{t-1} + b_i\right), \tag{3}$$

$$o_t = \sigma_g\left(W_o x_t + U_o c_{t-1} + b_o\right), \tag{4}$$

$$c_t = f_t \bullet c_{t-1} + i_t \bullet \sigma_c\left(W_c x_t + b_c\right), \tag{5}$$

$$h_t = o_t \bullet \sigma_h\left(c_t\right). \tag{6}$$

Equations (2)–(4) use the symbol *I*: update the gate, take sigmoid for the splicing result to indicate whether the previous result needs to be updated, *F*: forget the gate, take sigmoid for the splicing result to indicate whether the previous result is discarded and *O*: synchronize the gate, take sigmoid to indicate whether synchronization is required. *C* and *h*, which stand for cell states and hidden values, respectively, are used in equations (5) and (6). The forgetting gate, the input gate, and the output gate work together to calculate these two values. These gates have only few linear interactions with the other parts. σ is an activation function. The network using LSTM units provides excellent learning capabilities through modeling signal time feature, which yields the most advanced results in anomaly detection.

3.3. Fusion Model C^2-LSTM. C^2-LSTM uses a CNN and C-LSTM for fusion. For the purpose of extracting spatio-temporal features from the intrusion data center, CNN and C-LSTM are used as two independent strong learners. Anomaly detection is performed by splicing the two features through fusion. The two learners work parallelly to generate and evaluate the final result; or in other words, there is no reliance between them. Figure 2 illustrates its model structure.

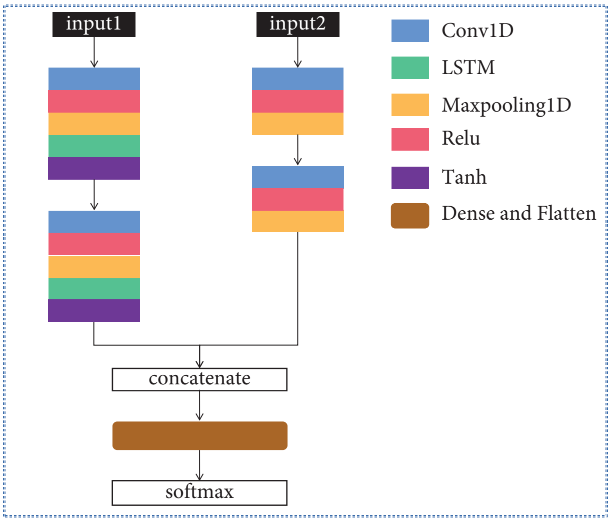


Figure 2: Model structure of our C^2-LSTM which consists of a CNN and our C-LSTM.

The features retrieved by the CNN and the C-LSTM are fused by fusion, and the features are stitched together after both the CNN and LSTM have derived their own values.

$$S_1 = \sigma\left(b_i + W_i \bullet x\right), \tag{7}$$

$$S_2 = o_t \bullet \sigma_h\left(c_t\right), \tag{8}$$

$$S = \left[\left[S_1\right], \left[S_2\right]\right], \tag{9}$$

where S_1 is the output of the C-LSTM, and S_2 is the output of the LSTM. *S* is the stitching of the two feature matrices in the last dimension. Here, we use the concatenate method to blend features and models. At the end of the model, a concatenate layer is built to combine the features extracted from the previous model. We spliced the tensors in the last dimension and ensured the alignment of the two parts of features in the last dimension. This makes the fused features rely on more than just the results of the previous operation step, combining features of different properties.

In traffic detection, the fully connected layer is responsible for reducing the sensitivity of the parameters in the learning process. And *Softmax* is used to output the final classification score. They are the layers used for the output of the C^2-LSTM model. In the upper part, the output of the fusion matrix is stretched, and this vector will be fed to the fully connected layer. This layer uses equation (10). *D* denotes the output of the fully connected layer, and σ is the activation function.

$$D_i = \sum \sigma\left(b_i + W_i \bullet h\right). \tag{10}$$

The output of the fully connected layer is multiclassified by *softmax* and *softmax* layer classifies the raw data into normal and abnormal classes.

3.4. Schema and Super Parameters. The input of C^2-LSTM is 41 ∗ 1 traffic input, and the parameters of various types of structures can be adjusted under the design conditions of the model. The input of the C2-LSTM is 41 ∗ 1 traffic data. Under

```
Input: $D_1 = \left\{X_1^1, X_1^2, \ldots, X_1^k\right\}, D_2 = \left\{X_2^1, X_2^2, \ldots, X_2^k\right\}$ are the input sets, and label $Y = \{y^1, y^2, \ldots, y^k\}$ is the corresponding
Output:A trained anomaly detection model $M$
Initialize the model $M$
Initialize the iteration count $T$, batch size $N$, threshold $\delta$
for $q = 1$ t o $T$ do
    for $m = 1$ to 2 do
        for each batch $\{X_m^i\}_{i=1}^N$ do
            Transfer $X_1^i$ into $S_1$ via CNN by equation (7)
            Transfer $X_2^i$ into $S_2$ via CNN by equation (8)
            Splice $S_1$ and $S_2$ into $S$
            Predict $y^{(i)}$ based on $Z$ via the estimation network
            Update $M$ to minimize loss
    end for
    if loss < $\delta$: break
end for
return $M$
```

Algorithm 1: Anomaly Detection Algorithm Based on C^2-LSTM.

the design conditions of the model, the parameters of various types of structures can be adjusted, such as the depth of CNN and LSTM, the design of convolution, and the gating strategy of LSTM. These settings will determine the final performance of the whole model, such as accuracy or learning speed. In contrast, the C^2-LSTM fusion method determines that it does not require a high number of layers. Before entering the LSTM, he becomes $19 * 32$ spatially feature-rich data by convolution layer and pooling layer. We use ReLU as the activation function of the model. After recent years of research, ReLU learns faster than Tanh, and he ensures that the product of the parts is all always 1, and there is no problem of gradient disappearance of the sigmoid.

3.5. C2-LSTM Anomaly Detection Algorithm. The workflow of C^2-LSTM anomaly detection is as follows: first, preprocess and normalize sample raw data to handle dirty data and construct input sets X1 and X2. Then, the input datasets are learned with CNN and C-LSTM, respectively, and the outputs of both are two feature matrices S_1 and S_2. By dimensional stitching of feature fusion, S_1 and S_2 are reconstructed into S. The reconstructed features S will be used in the final prediction to get the final classification of that traffic. The specific algorithm is shown in Algorithm 1.

4. Experiment and Analysis

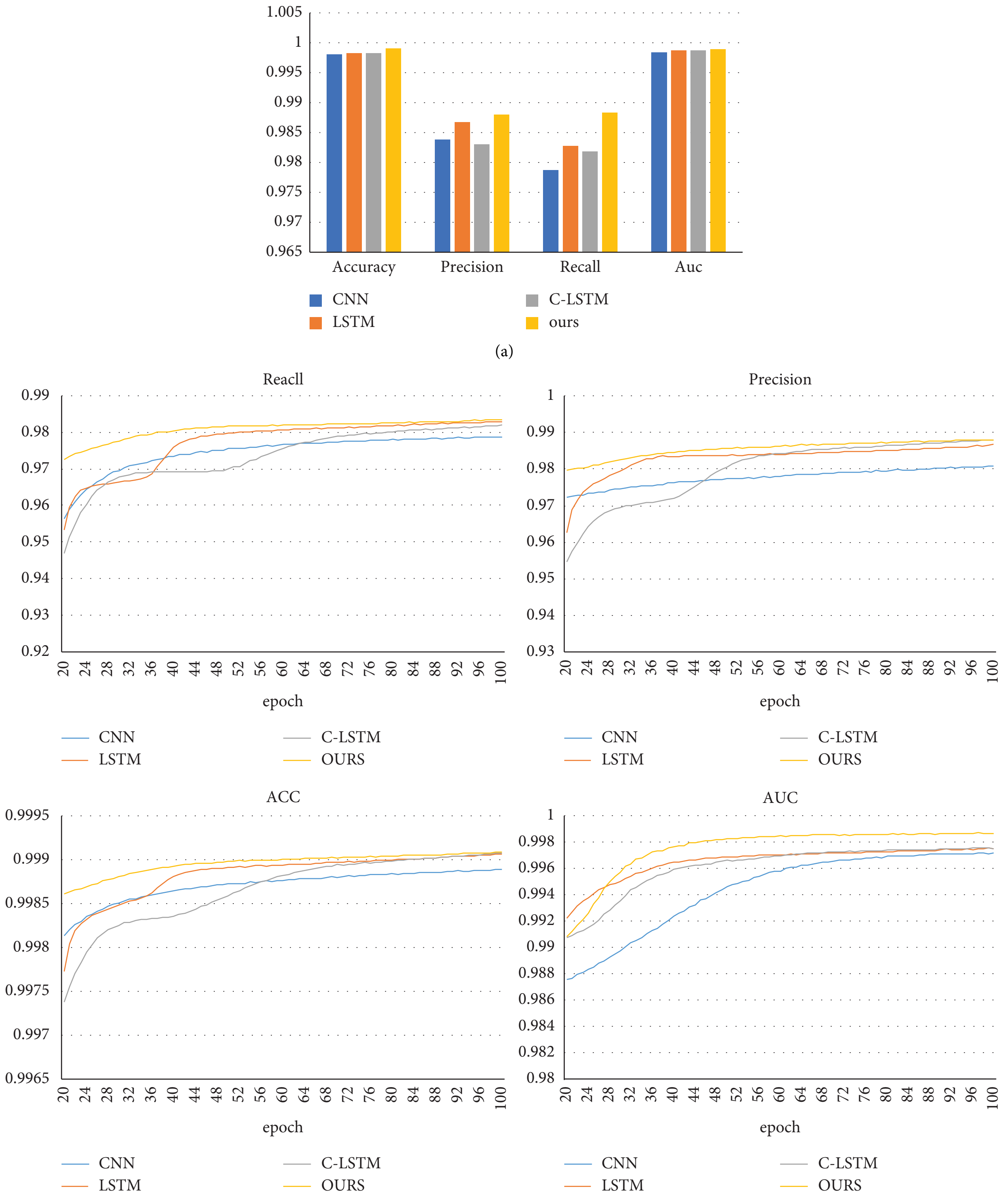
We use Python 3.7 as the programming language and the CUDA version of Tensorflow 1.14.0 as the neural network framework. The lab deploys comparison experiments in the same environment and trains models on 4 GeForce RTX 2080 Ti with 12G video memory. We use small batch training in the experiments with a batch size of 512. The details of the experiments are as follows.

4.1. Data Sets and Preprocessing. KDD99 is a dataset for monitoring abnormal connections from normal connections, from the DARPA Intrusion Detection Evaluation Project in 1998. The KDD99 dataset is a feature extract version of the DARPA dataset (DARPA is the original dataset), and the training data for the experiment were 7 weeks of network traffic. This dataset was utilized in the KDDCUP competition in 1999 and later became known as the KDD99 dataset. Although the dataset is too old and may have obsolescence issues, KDD99 was very popular among researchers and sets the stage for deep learning and intelligent computing to make a big splash in intrusion research.

Each entry in the dataset is labeled, specifically into 2 types of anomalous attacks and 1 type of normal. We train a random sample at a time to learn the characteristics of each anomaly type in order to make predictions for each input data.

In the experimental study, the network intrusion detection packet kdd_cup_data_10percent from KDDCup99 is marked as the training set and corrected as the test set. The kddcup_data_10percent packet is a 10% sample of the kddcup_data packet. Since the data processed for the experiment is network traffic, inputting a segment of network traffic predicts the category to which it belongs (39 attacks + normal). For such a classification problem, we conducted similar experiments in different models and evaluated these models by accuracy, precision, recall, and f1 score.

To explain our evaluation metrics, the following explanation is given. Suppose a correct sample is incorrectly considered as wrong in a dichotomous classification problem, and this wrong data is labeled as false positive (FP). A false negative (FN) indicates that an abnormal instance is labeled as normal. Similarly, true positives (TP) and true negatives (TN) indicate abnormalities and correctly identify normal instances. The area enclosed by axes under the ROC curve is defined as the AUC (area under the curve), which has values between 0.5 and 1 in a $1 * 1$ coordinate system. The closer the AUC is to 1.0, the better the prediction equals to 0.5, the lowest truthfulness and no application value. Different metrics can be evaluated in this way:



(a)

(b)

Figure 3: (a) Comparison of the metrics on the test data set and (b) comparison of the metrics on 100 epochs during training.

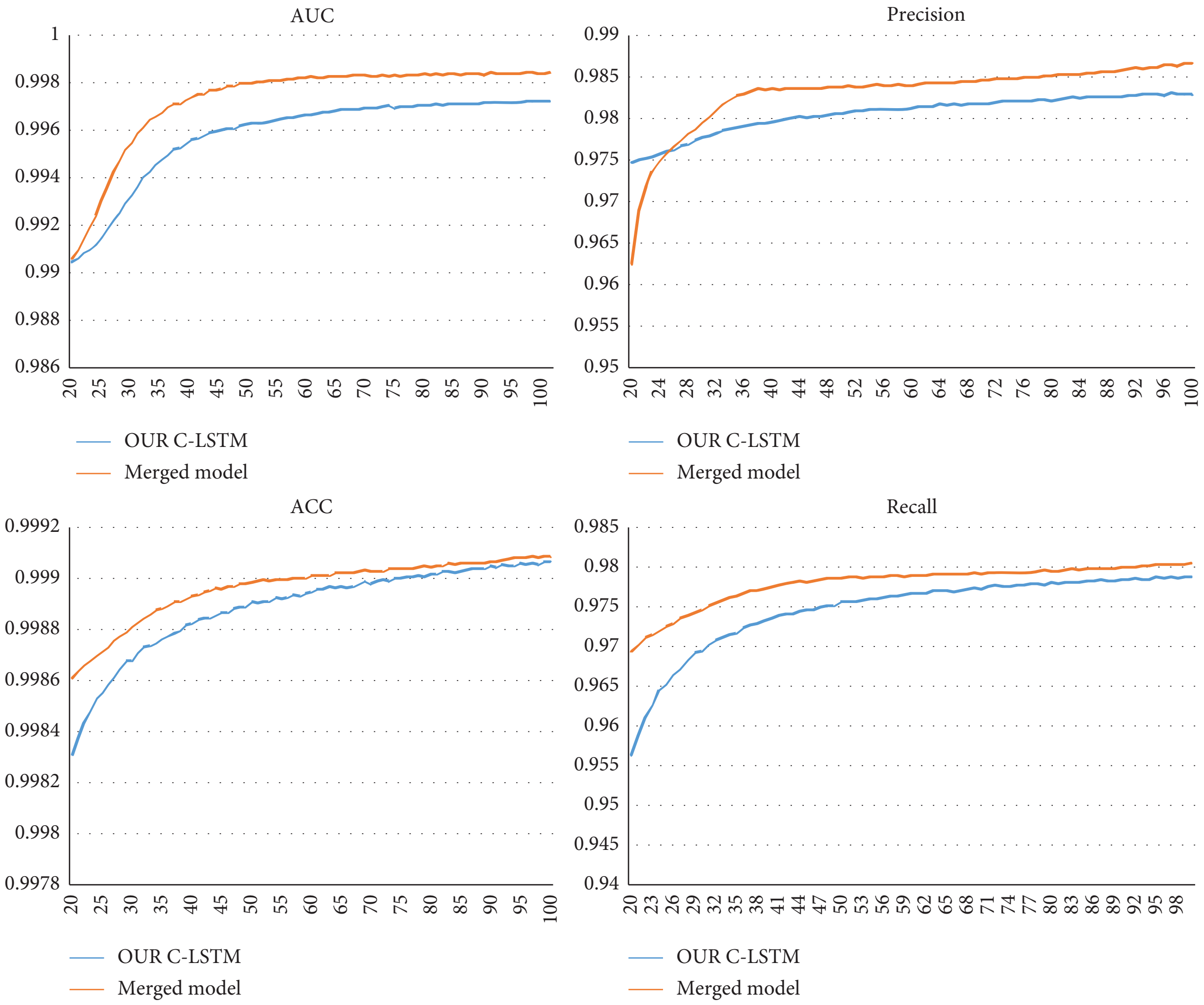


Figure 4: Train these two models with 100 epochs of training data for comparison.

$$
\begin{aligned}
\text{accuracy} &= \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}},\\
\text{precision} &= \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}},\\
\text{recall} &= \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}},\\
\mathrm{AUC} &= \frac{1}{2}\sum_{i=1}^{m-1}\left(x_{i+1} - x_i\right)\bullet\left(y_{i+1} + y_i\right).
\end{aligned}
\tag{11}
$$

Data preprocessing using python consists of numerical replacement text, numerical normalization, and tag unique hot coding. Numerical replacement text mainly converts the values of the 41 feature values of each connection that are strings into numerical form. The most-valued normalization is used in numerical normalization. The preprocessing ends into 4 files (train_x, train_y, test_x, test_y).

4.2. Analysis of Indicators. To build our model and conduct experiments, we utilized two models; the CNN can extract spatial features up to dimensionality, while the C-LSTM is more sensitive to changes on time steps. We designed experiments to evaluate the effect of CNN and C-LSTM fusion. We set the number of convolution cores of the convolution layer to 32 and the step size to 3. The window size of the pooled layer is 2. The number of LSTM cells is 64. In this experiment, we evaluated four models including CNN, LSTM, C-LSTM, and C^2-LSTM. After each model was trained on the training dataset for 100 calendar hours at a learning rate of 1e-2, the performance (precision, accuracy, recall, and Auc score) on the test dataset was collected as shown in Figure 3. The fusion model with C^2-LSTM has the highest scores in terms of training accuracy, precision, recall, and Auc. From the test results, we can conclude that our C-LSTM outperforms the single CNN and LSTM. Proving that it can extract more key features in time series, our C-LSTM also demonstrates better temporal feature extraction and is higher than the original LSTM in terms of AUC, accuracy, and precision metrics compared to the single LSTM. At the same time, it is slightly lower than the LSTM in terms of recall metrics, which we believe is due to the one-dimensional CNN in front of the model that makes it focus more on single temporal features and ignore spatial features in some dimensions.

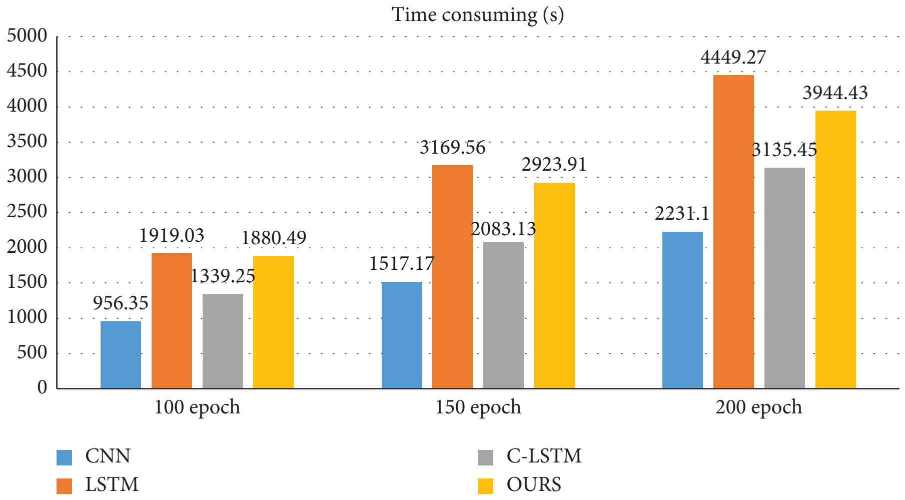


Figure 5: Convergence speed of training.

4.3. Comparison with Our C-LSTM. In the paper [7], the authors use sliding windows to construct preprocessed temporal correlation data and use LSTM to extract temporal features. In order to achieve better results in temporal feature extraction with the fused model, our improved model replaces the C-LSTM's two-dimensional convolution and sliding window operation for extracting temporal features with a one-dimensional temporal convolution and fuses it with the CNN. To demonstrate the effectiveness of the improvement and fusion, we take out the improved model separately and compare it with the C^2-LSTM. We evaluate the classification results of the C-LSTM model with one-dimensional convolution and the C^2-LSTM. As shown in Figure 4, the two models are trained with 100 epochs of training data with a learning rate of 1e-3. The lowest loss values and best performance across all metrics are obtained by the fused C^2-LSTM. The fused C^2-LSTM based on fusion can extract superior features for further classification, according to the experiments.

4.4. Time Performance Requirements. Driven by the new digital revolution represented by IoT technologies, these emerging resource-constrained network entities usually have limited computing power or more sensitive latency characteristics. Therefore, the training and detection time of the network and its own lightweight are also discussed in our consideration and comparison experiments. We compared the training time and prediction time for CNN, LSTM, our modified C-LSTM, and our fused C^2-LSTM. The experimental conditions were a GeForce RTX 2080 Ti with 12G video memory. The training time is the total training time when training 100 epochs with a batch size of 512, and the prediction time is the overall prediction time when predicting 494021 data using the trained model. Figure 5 depicts the performance at training time, with the fused C^2-LSTM converging fastest at a lower loss. Our C^2-LSTM offers faster prediction and training speed compared to simple CNN, as seen in Table 3. This is due to the fact that our fusion model possesses superior capability in prediction without requiring deep network layers.

Table 3: Time consumption for training and prediction.

	CNN (*s*)	LSTM (*s*)	Our C-LSTM (s)	Ours (s)
Training time	956.35	1919.03	1339.25	1880.49
Testing time	3.06	3.09	2.81	2.62

5. Conclusions

We have proposed a new architecture combining CNN and enhanced C-LSTM to better adapt to the emerging IoT anomaly intrusion detection with massive data and high latency sensitivity. In addition, we demonstrate that the architecture that extracts spatial and temporal features separately in parallel from CNN and C-LSTM before fusing them can better learn both spatial and temporal correlations of data simultaneously to better cope with complex IoT environments. Based on this, we have evaluated different anomaly detection methods and used C^2-LSTM to extract superior features for classification in fully connected networks. According to the results of the experiments, the model has performed at the highest level in terms of accuracy, precision, completeness and AUC score. Furthermore, its model structure determines that it can boost detection performance without a deep network and can also evaluate temporal performance at a higher level. It is challenging to sustain its existing edge over shallow networks in the face of ultrahigh latitude data, though, as the complexity of the data keeps growing. When faced with such data, we have intended to use PCA to downscale and process the data, but using data preprocessing methods will inevitably introduce some latency, which is not permitted in industrial IoT devices listed with high latency sensitivity, and we will continue to work in this direction in the future.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Key R&D Program of China (no. 2022YFB3104100), the Major Research Plan of the National Natural Science Foundation of China (no. 92167203), the Guangzhou Science and Technology Plan Project (no. 202201020216), and the Guangxi Key Laboratory of Cryptography and Information Security (no. GXIS202119).

References

[1] R. Vinayakumar, M. Alazab, S. Srinivasan, Q. V. Pham, S. K. Padannayil, and K. Simran, "A visualized botnet detection system based deep learning for the internet of things networks of smart cities," *IEEE Transactions on Industry Applications*, vol. 56, pp. 4436–4456, 2020.

[2] S. Smith, "IoT connections to reach 83 billion by 2024, driven by maturing industrial use cases," 2020, https://www.juniperresearch.com/press/iot-connections-to-reach-83-bn-by-2024#:%7E:text=Industrial%20Use%20Cases-,IoT%20Connections%20to%20Reach%2083%20Billion%20by%202024%2C%20Driven%20by,35%20billion%20connections%20in%202020.

[3] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.

[4] E. Schubert, A. Zimek, and H. P. Kriegel, "Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection," *Data Mining and Knowledge Discovery*, vol. 28, pp. 190–237, 2014.

[5] M. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proceedings of the Computer Vision-ECCV 2014 13th European Conference*, Zurich, Switzerland, September 2014.

[6] B. Lindemann, B. Maschler, N. Sahlab, and M. Weyrich, "A survey on anomaly detection for technical systems using LSTM networks," *Computers in Industry*, vol. 131, Article ID 103498, 2021.

[7] T. Y. Kim and S. B. Cho, "Web traffic anomaly detection using C-LSTM neural networks," *Expert Systems With Applications*, vol. 106, pp. 66–76, 2018.

[8] A. Preciado-Grijalva and V. R. Iza-Teran, "Anomaly detection of wind turbine time series using variational recurrent autoencoders," 2021, https://arxiv.org/abs/2112.02468.

[9] P. Malhotra, L. Vig, and G. Shroff, "Long short-term memory networks for anomaly detection in time series," in *Proceedings of the 31th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, vol. 89, Bruges, Belgium, December 2015.

[10] P. Malhotra, A. Ramakrishnan, and G. Anand, "LSTM-based encoder-decoder for multi-sensor anomaly detection," 2016, https://arxiv.org/abs/1607.00148.

[11] D. M. Hawkins, *Identification of Outliers*, Chapman and Hall, London, UK, 1980.

[12] B. Abraham and A. Chuang, "Outlier detection and time series modeling," *Technometrics*, vol. 31, pp. 241–248, 1989.

[13] M. Markou and S. Singh, "Novelty detection: a review—part 2: statistical approaches," *Signal Processing*, vol. 83, pp. 2499–2521, 2003.

[14] M. Markou and S. Singh, "Novelty detection: a review—part 1: statistical approaches," *Signal Processing*, vol. 83, pp. 2481–2497, 2003.

[15] R. Stephen and L. Arockiam, "Intrusion detection system to detect sinkhole attack on RPL protocol in Internet of Things," *International Journal of Electrical, Electronics and Computer Systems*, vol. 4, pp. 16–20, 2017.

[16] G. Münz, S. Li, and G. Carle, "Traffic anomaly detection using k-means clustering," in *Proceedings of the 2016 3rd International Conference on Recent Advances in Information Technology (RAIT)*, vol. 7, Dhanbad, India, March 2007.

[17] J. Zhang and M. Zulkernine, "Anomaly based network intrusion detection with unsupervised outlier detection," in *Proceedings of the 2006 IEEE International Conference on Communications*, vol. 5, IEEE, Istanbul, Turkey, June 2006.

[18] G. Kaur, A. H. Lashkari, and A. Rahali, "Intrusion traffic detection and characterization using deep image learning," in *Proceedings of the 19th IEEE International Conference on Dependable, Autonomic & Secure Computing (DASC 2021)*, Calgary, Canada, June 2020.

[19] M. M. Hassan, A. Gumaei, A. Alsanad, M. Alrubaian, and G. Fortino, "A hybrid deep learning model for efficient intrusion detection in big data environment," *Information Sciences*, vol. 513, pp. 386–396, 2020.

[20] L. Zhu and N. Laptev, "Deep and confident prediction for time series at uber," in *Proceedings of the 2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, IEEE, New Orleans, LA, USA, November 2017.

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

[22] V. Kindl, B. Skala, R. Pechanek, V. Kus, and J. Hornak, "Low-pass filter for HV partial discharge testing," *Sensors*, vol. 18, p. 482, 2018.

[23] Y. Tian, M. Mirzabagheri, S. M. H. Bamakan, H. Wang, and Q. Qu, "Ramp loss one-class support vector machine; a robust and effective approach to anomaly detection problems," *Neurocomputing*, vol. 310, pp. 223–235, 2018.

[24] J. Takeuchi and K. Yamanishi, "A unifying framework for detecting outliers and change points from time series," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, pp. 482–492, 2006.

[25] J. Zhu, Y. Wang, D. Zhou, and F. Gao, "Batch process modeling and monitoring with local outlier factor," *IEEE Transactions On Control Systems Technology*, vol. 27, pp. 1552–1565, 2019.

[26] L. E. Nugroho, L. Lazuardi, and A. S. Prabuwono, "Detection of anomalous vital sign of elderly using hybrid k-means clustering and isolation forest," in *Proceedings of the TENCON 2018-2018 IEEE Region 10 Conference*, IEEE, Jeju, Korea, October 2018.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 6546004, 13 pages
https://doi.org/10.1155/2022/6546004

Research Article

IoV-SDCM: An IoV Secure Data Communication Model Based on Network Encoding and Relay Collaboration

Yan Sun,[1] **Lihua Yin**,[1] **Ying Ma**,[2] **and Chonghua Wang**[3]

[1]*Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou, China*
[2]*State Information Center, Beijing, China*
[3]*China Industrial Control Systems Cyber Emergency Response Team, Beijing, China*

Correspondence should be addressed to Lihua Yin; yinlh@gzhu.edu.cn and Chonghua Wang; chonghuaw@live.com

Received 25 July 2022; Revised 26 September 2022; Accepted 11 October 2022; Published 26 November 2022

Academic Editor: Xiaofan Liu

Internet of Vehicles (IoV) is a significant 5G application scenario. As it developed rapidly, more and more vehicles are connected to Internet of Vehicles. The data security and privacy are the premises to ensure its service quality in an open communication network. This paper proposes IoV-SDCM, a secure data communication model in IoV. It includes a self-organizing relay forwarding network and an assured delivery mechanism. The relay forwarding network is used for constructing a dynamic collaboration network with the vehicle as the node. The security delivery mechanism is that network coding is used for data fragmentation and re-encoding to improve network communication reliability. Homomorphic encryption is used to encrypt and protect the encoding vector, improving information leakage and anticollusion attacks. The theoretical proof proves that the model has the ability of data transmission confidentiality and better antiattack capabilities, while it has privacy protection capabilities. Furthermore, the experiment verifies that the model also has the advantage of high and stable data delivery efficiency.

1. Introduction

Internet of Vehicles (IoV) is a new form of mobile Ad hoc networks (MANETs) in the field of road traffic, which is an intelligent information network service [1]. Vehicles are connected, and real-time information exchange is performed on roadside facilities, the Internet, and transportation systems in an open and joint environment [2]. However, the rapid movement of IoV nodes, dynamic topology changes, and on-demand connectivity challenge security and privacy [3]. Its security can be roughly divided into security of terminals, communication networks, and cloud platforms. Especially in the IoV communication process, the high openness is well used to build a transmission network. However, with the high-speed mobility of vehicles, the network topology changes dynamically, and the quality of data transmission service is difficult to guarantee. At the same time, fake nodes or malicious attack entities will also be introduced to launch Sybil attacks, DDoS, and APT attacks. Or infer some privacy information for identity, location, preferences, motion trajectory by traffic analysis, packet analysis and tracing, and collusion attacks. These issues cause that the vehicle control signals and alarm signals are unable to be transmitted to the vehicle terminal in a timely and fast manner or are intercepted and tampered by attackers. They may even greatly impact the safety of users' personal and property [4, 5]. There is a further problem with the effective combination and balance of communication efficiency and security.

To solve the communication efficiency problem, some researchers study V2V relay communication strategies based on node self-organizing cooperation to improve the throughput of in-vehicle networks. For collaborative data distribution and transmission, most researchers use content relay and perfecting schemes [6], and some researchers use game theory-based incentive mechanisms [7, 8] or some learning algorithm [9] to promote cooperation between nodes. These methods have made contributions to the transmission delay and transmission flow and have a significant role in promoting vehicle data transmission in high-speed road scenarios or urban road scenarios. At the same time, many researchers have paid great attention to the above security issues in the new scenarios of 5G IoV communication and some researchers have also focused on the effective combination and balance of communication efficiency and security. The concept of network coding was proposed in 2000 [10]. On the one hand, network coding technology can improve the robustness of the network, resist the impact on network links, reduce retransmissions, and reduce network management overhead. On the other hand, it can improve the security of information. Through the XOR operation, it is equivalent to encrypting the information, making the information more difficult to be eavesdropped. Even if the information is eavesdropped, it is difficult for the eavesdropper to decode the information correctly because he does not know the processing method of the information and cannot obtain valid information. Therefore, some researchers have used it for vehicle network data communication [11–13] and secure communication [14, 15]. Meanwhile, some researchers use anonymity, encryption, and one or more technologies to design IoV privacy protection mechanisms for solving node identity privacy, location privacy, and data privacy leakage during communication [16, 17].

After research and analysis, it is necessary that a secure data communication scheme of IoV is constructed, configuring a relay collaboration strategy to improve transmission performance and a security policy to guarantee security and privacy. This paper proposes IoV-SDCM, a secure data communication model in IoV. Our main contributions include the following:

(1) We design a self-organizing data communication network composed of relay cooperative vehicle nodes. Through a pseudonymous strategy and a broadcast policy, it achieves source node anonymity, target node anonymity, and communication relationship anonymity in the self-organizing relay forwarding network.

(2) A data delivery strategy is proposed utilizing random network coding aided by homomorphic encryption in the data communication. It increases network throughput while protecting the confidentiality of information. And it can defend against collusion attacks during network coding transmission as the global network coding vectors are encrypted by homomorphic encryption.

(3) After the theoretical proof and performance analysis, the proposed model reduces the overhead of encryption and decryption and the computing cost of nodes. Furthermore, it ensures the confidentiality of data transmission and privacy protection capabilities. We conclude that the model has high reliability and good performance by simulation experiments.

The rest of the paper is organized as follows. Section 2 provides the related work. The proposed IoV-SDCM model is described in Section 3 and the theoretical proof is analyzed in Section 4. Section 5 reports the experiments in detail and discusses the experimental results. Finally, a brief conclusion is drawn in Section 6.

2. Related Work

2.1. Network Coding. Initially, some researchers have made some progress in using network coding to improve the communication performance of the Internet of Vehicles. Ahlswede et al. [10] proposed the concept of network coding in 2000. Ho et al. [18, 19] proposed an algorithm of random network coding (RNC). It is simple in construction and easy to implement in relay collaborative IoV communication. Some scholars have used network encoding to provide IoV data transmission and improve the stability and security of data transmission between dynamic nodes. Kai et al. [20] proposed an auxiliary scheduling algorithm based on network coding to achieve data sharing and collaboration between V2X, which improved data services' performance and bandwidth efficiency and reduced the risk of direct data exposure. Kwon and Park [21] proposed a V2I real-time data distribution system for system network encoding, aiming at the validity and reliability of V2I data transmission. It could effectively reduce the network delay in V2I communication caused by packet loss in the channel. Gao et al. [22] proposed a network coding system that assists D2D transmission, which improved the total network capacity using a payoff function balancing relay selection and resource allocation under complex interference conditions. The above research could effectively improve the throughput of the network, but no further research has been conducted on possible security risks.

Next, some researchers attempt to solve the transmission performance and some security problems by network coding. Khan and Chatzigeorgiou [23] proposed an opportunistic relay framework based on random network coding. It simulated the probability that it could partially or wholly recover confidential data if an eavesdropper intercepted a certain number of packets. And it also validated the trade-offs between security and reliability. Xu et al. [24] proposed a transmission scheme using adaptive relay selection, in which users promote secure communication through collaboration. It had a stable performance gain and effectively suppressed eavesdropping channels. However, due to its security problems by itself, it is still unable to solve the security problems such as conspiracy attacks in data transmission.

2.2. Privacy Protection. At the same time, some researchers have designed the IoV privacy protection using one or more technologies such as anonymity and encryption. Kang et al. [25] proposed that IoV edge resources and fog computing technology can effectively manage and distribute pseudonyms for identity authentication. It improves the ability of identity privacy protection. Wang et al. [26] designed a binary privacy-preserving scheme. The scheme used decentralized CA and biometric password-based authentication to reduce authentication costs and achieve conditional privacy protection. Rajput et al. [27] designed a hierarchical pseudonym authentication protocol that relied solely on CA no longer and reduced the burden on IoV systems. Rabieh et al. [28] gave a route privacy protection method using homomorphic encryption and error-checking technology, which protects the driver's trajectory data privacy and prevents collusive attacks between malicious vehicles. The above research has only improved in security and privacy, but limited improvement in network performance.

2.3. Our Motivation. Our paper focuses on a model for secure data communication based on network coding and relay collaboration. The data communication network is constructed based on relay collaboration vehicle nodes. In the relay collaborative communication network, the relay node could expand communication coverage, effectively improving communication quality and increasing the eavesdropped information risk. We deeply study the stability and security of data transmission and give the corresponding mechanism for security and privacy protection.

3. IoV-SDCM

Section 3.1 gives IoV-SDCM's model definition and components. For the detailed components, the relay forwarding network and security data delivery mechanism are described in Section 3.2 and Section 3.3.

3.1. Model Definition

Definition 1. IoV secure data communication model (IoV-SDCM): IoV-SDCM could be defined as quadruples (S, D, N, T), where S is the source vehicle node, D is the target vehicle node, N is the forwarding network including relay cooperative vehicle nodes, and T is the secure transmission policy, as shown in Figure 1.

(i) Source vehicle node S: The source vehicle node S divides the information into m slices and uses the pseudonymous strategy to generate m virtual source vehicle nodes, each of which has a data fragment.

(ii) Target vehicle node D: target vehicle node D receives the encoding information, global encoding vectors, and local encoding matrix from the last relay vehicle nodes and decodes to restore the original data.

(iii) Forwarding network N: The forwarding network is defined as a wireless multihop network composed of $m \times n$ relay nodes $R_{(i,j)}$ ($i = 1,2, \ldots, m; j = 1,2, \ldots, n$; $R_{(i,j)}$), where $R_{(i,j)}$ means the j node in the ith hop group. The nodes and their links in the forwarding network meet the following properties at the same time: (1) there are m neighbour nodes within a hop of source node S as the entry nodes $R_{(i,j)}$ ($i = 1,2, \ldots, m; j = 1$); (2) there are m neighbour nodes within a hop of target vehicle node D as the exit nodes $R_{(i,j)}$ ($i = 1,2, \ldots, m; j = n$); (3) the relay nodes within the adjacent one-hop range are all within the communication range of each other; (4) the length of each path is n; (5) there are m disjointed data forwarding paths from the entry node to the exit node.

(iv) Secure transmission policy T: The source node uses the homomorphic encryption function to encrypt the initial global encoding vector and uses a random coefficient to encode the information slices for network encoding, which are transmitted to the entrance nodes, respectively. The relay node encodes the data slices by random coefficient selection. It uses the splitting forwarding strategy in the anonymous forwarding network to transmit the encoding information, global encoding vectors, and local encoding matrix. Finally, the exit node broadcasts encoding information, a global encoding vector, and a local encoding matrix to the target node.

3.2. Relay Forwarding Network. It is necessary to meet the requirements of source node anonymity, target node anonymity, and communication relationship anonymity in the self-organizing relay forwarding network. We use the pseudonymous strategy for generating multiple virtual source nodes to achieve the anonymity of the source node. We build an anonymous relay forwarding network to achieve anonymous communication relationships. In the anonymous relay forwarding network, each hop contains a group of nodes, the groups can communicate with each other, and each group of nodes only knows its previous hop group and the next hop group. The exit node broadcasts information to the target node to achieve receiver anonymity. The specific forwarding network construction process and related anonymity strategies are shown in Figure 2.

3.2.1. Initialization. First, a forwarding link from a source vehicle node to the target vehicle node is generated.

The source vehicle node S routs a request message $RREQ$ to target vehicle node D. The structure of $RREQ$ is shown in Table 1. If the adjacent node is not the target vehicle node, it is logged to the $RREQ$ message, and the number of paths is increased by 1. Then, it continues to be forwarded. Otherwise, it stops forwarding and gets a forwarding path from the source vehicle node S to target vehicle node D if an adjacent node is the target vehicle node. Finally, the target vehicle node D sends an answer message containing the path information $path$ to route request node S.

3.2.2. Source Vehicle Node Anonymity Policy. A virtual source node strategy is proposed to achieve the source vehicle node S anonymity. It generates m virtual source nodes as forwarding nodes using the pseudonymous mechanism and generates m forwarding paths.

Supposing the source vehicle node S is identified as ID_s, S presets a hash function H and a random number generator that results in a random number α_i, where ID_s and H are l bits. Finally, S uses the hash function to generate the pseudonym set $S' = \{S_1, S_2, \ldots, S_m\}$. The pseudonym of the source node is

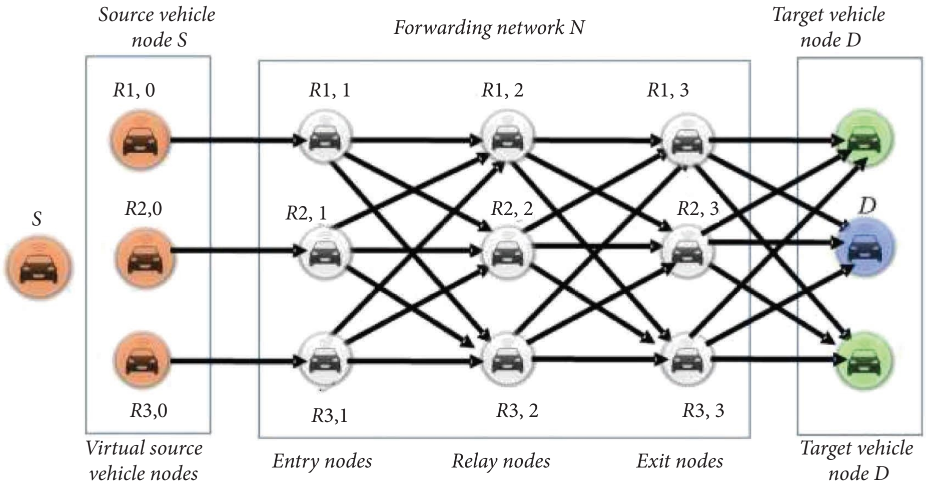


Figure 1: IoV data security delivery model.

$$S_1 = H\left(ID_S \oplus \alpha_1\right),$$
$$S_2 = H\left(ID_S \oplus \alpha_2\right), \quad (1)$$
$$S_m = H\left(ID_S \oplus \alpha_m\right).$$

3.2.3. Relay Node Anonymity Policy. In a routing path generated by initialization from source vehicle node S to target node D, node i as the group header ($Header_i$) is selected by neighbouring nodes, constructing the ith hop m-anonymous group (Hop_Group_i). And its member node g_j meets the following conditions:

(i) The ith hop m-anonymous group Hop_Group_i is within the communication scope of its previous anonymous group ($Hop\_Group_{i\text{-}1}$) and the next hop anonymous group (Hop_Group_{i+1})

(ii) The group head node $Header_i$ could set the forwarding path sequence g_j for the member nodes in the group, and the forwarding path sequence of the member nodes is shown in Table 2.

3.2.4. Exit Node Broadcast Policy. For the exit node ($hop = n$), a set of forwarding nodes containing m nodes is generated. In the communication range of the previous hop forwarding node, there are a developed set of forwarding nodes. The target node D, i.e., ID_R, is within the broadcast range of the exit node set.

3.2.5. Update Policy. We set the communication cycle T, in which the source vehicle node carries data transmission along the constructed anonymous forwarding network. When the next communication cycle arrives, the original anonymous forwarding network is abandoned, and an anonymous forwarding network is re-established for data transmission. It could prevent the failure of the routing node and balance the energy consumption.

The specific process of constructing a self-organizing anonymous forwarding network is shown in Algorithm 1.

3.3. Secure Data Delivery Mechanisms. As shown in Figure 3, we suppose a trusted authority distributes a key pair (k_e, k_d) for each node, where k_e is an encryption key, and k_d is a decryption key, and the encryption key k_e is issued to all other nodes.

Phase 1. The source vehicle node S divides the information to be sent M into m slices of information ($M_1, M_2, \ldots, M_m$). Taking $m = 3$ as an example, it generates m virtual nodes, assigning the encoded fragments to the virtual nodes.

If the source vehicle node does not know the target vehicle node key, we preprocess the original data using the information-slicing strategy. This mechanism aims to ensure the confidentiality of the data during transmission. A specific method of slicing information is given as follows.

An original message M of the source vehicle node is sliced into m data fragment. The length of a data fragment is d, and then, the original message M can be represented as an m-dimensional vector:

$$M = \left(M_1, M_2, \ldots M_m\right). \quad (2)$$

Since plaintext transmission of shared information leaks content to relay nodes in the forwarding network, plaintext transmission is not ideal. By introducing a random but reversible transformation matrix A to construct a perturbation source information slice, the original information after the disturbance M' is as follows.

$$\begin{aligned} M' &= AM \\ &= \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{pmatrix} \left(M_1, M_2 \ldots, M_n\right) \\ &= \left(A_1 M_1, A_2 M_2, \ldots, A_n M_n\right) \\ &= \left(M_2', M_2', \ldots, M_n'\right). \end{aligned} \quad (3)$$

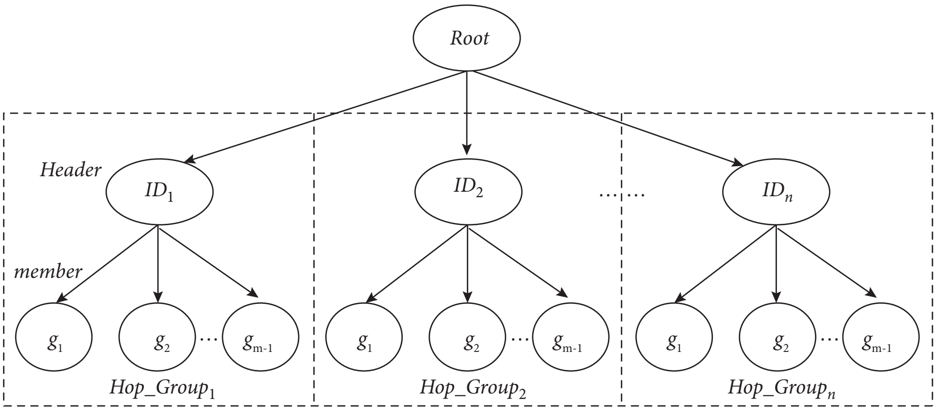


FIGURE 2: A self-organizing anonymous forwarding network based on groups.

TABLE 1: Routing request message RREQ.

Source	Target	*Num*	path [1]	path [2]	......	path [n]
ID_S	ID_R	n	ID_1	ID_2	......	ID_n

Thus, the source information transmitted in the anonymous forwarding network is the information M transformed information.

$$M' = \left(M_1', M_2', \ldots M_m'\right). \tag{4}$$

The information-slicing policy avoids the direct transmission from leaking content to relay nodes in the forwarding network.

For the target vehicle node, as long as all slicing information of M' and transformation matrix A are received, the original information M can be restored, i.e.,

$$\begin{aligned} M &= A^{-1}M' \\ &= A^{-1}\left(M_1', M_2', \ldots M_m'\right). \end{aligned} \tag{5}$$

Phase 2. The virtual source node encodes each slicing information separately and then sends the encoded data and the encrypted global encoding vector to the entry node, respectively.

The source node S builds m different forwarding paths for the data slices. For the original data M, the source node divides it into m information slices (M_1, M_2, . . ., M_m) in the source node data slicing strategy. Its encoding forwarding strategy is that the source node selects a random coefficient for each slicing data and computes the network encoding, and then, it sends the encoded data and the encrypted global encoding vector to the next hop node along m different paths.

Assuming the length of each slice data d, the specific steps for the source node data encoding are as follows:

Step 1. Random coefficient selection for network encoding.

Randomly select m coding coefficient vectors of length d on a finite field F, ${C_{(i,0)}}^j$ (i = 1,2, . . ., m; j = 1,2, . . ., m), which forms a local encoding matrix, denoted $C_{(i,0)}$ (i = 1,2, . . ., m), i.e.,

TABLE 2: The m-anonymous group of the ith hop (Hop_Group$_i$).

hop	hop$_1$	hop$_2$	......	hop$_j$	......	hop$_m$
i	g_1	g_2		g_j		g_m

$$C(i,0) = \begin{bmatrix} C_{(i,0)}^1 \\ C_{(i,0)}^2 \\ \vdots \\ C_{(i,0)}^m \end{bmatrix} (i = 1, 2, \ldots, m). \tag{6}$$

Step 2. Network encoding for each slice M_i.

The local encoding matrix $C_{(i,0)}$ and each slice M_i operate a binary bit addition. That is, each encoding coefficient vector of $C_{(i,0)}$, ${C_{(i,0)}}^j$ performs an XOR operation on M_i, separately, denoted as follows:

$$\begin{aligned} M_i^{(0)} &= C_{(i,0)} \oplus M_i \\ &= \left(C_{(i,0)}^1 \oplus M_i, C_{(i,0)}^2 \oplus M_i, \ldots, C_{(i,0)}^m \oplus M_i\right) (i = 1, 2, \ldots, m). \end{aligned} \tag{7}$$

Step 3. Calculating the global encoding vector.

The global coding vector ${V_i}^{(0)}$ consists of the ith coding coefficient component of $C_{(i,0)}$, and then, the global coding vector is as follows:

$$V_i^{(0)} = \left(C_{(1,0)}^i, C_{(2,0)}^i, \ldots, C_{(m,0)}^i\right). \tag{8}$$

Step 4. Encrypting the global encoding vector.

The global encoding vector ${V_i}^{(0)}$ is encrypted using a homomorphic cryptographic function, denoted as

$$\begin{aligned} EV_i^{(0)} &= E_H\left(V_i^{(0)}, k_e\right) \\ &= E_H\left(\left(C_{(1,0)}^i, C_{(2,0)}^i, \ldots, C_{(m,0)}^i\right), k_e\right) \\ &= \left(EC_{(1,0)}^i, EC_{(2,0)}^i, \ldots, EC_{(m,0)}^i\right). \end{aligned} \tag{9}$$

Step 5. Forwarding the encoded message.

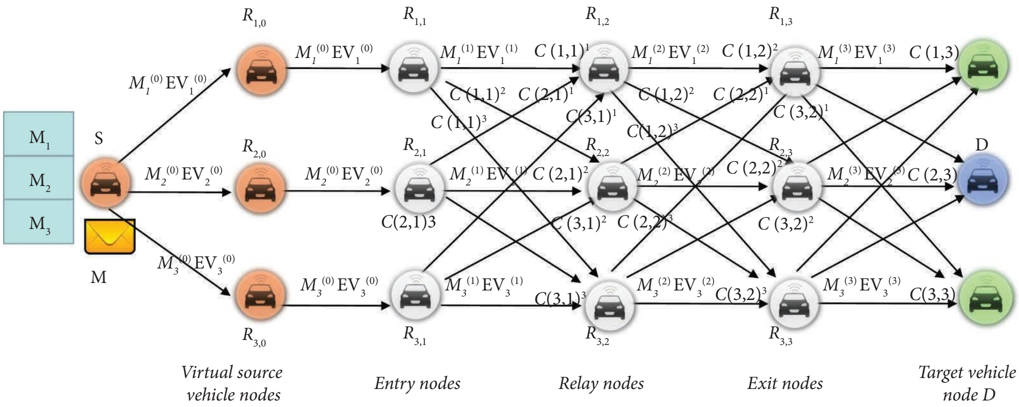


Figure 3: Secure transmission mechanism ($m = 3$, $n = 3$ as an example).

The source node sends the encoded information $M_i^{(0)}$ and the encrypted global encoded vector $EV_i^{(0)}$ along the ith path to the exit node.

The above encoding forwarding process avoids the direct decoding of a single entry node because the encoded data fragments consist of the splicing data and column vector of the local encoding matrix. The encoded data slices and encrypted global encoded vectors are forwarded along the m-path, respectively, preventing the exit nodes from colluding to recover the original data.

Phase 3. The entry nodes re-encode and use the splitting forwarding strategy for transmission after receiving the encoded information and the encrypted global coding vector. The jth relay node of the ith hop $R_{(i,j)}$ receives the m packets sent by the relay node of the previous hop m-path, selects the random coefficient, and encodes m packets. Then, according to the list of neighbour nodes $R_{(i,j)}$, the encoded information, the global encoding vector ciphertext encrypted, and the row vector of the local encoding vector are forwarded to the next hop relay node along m different paths. Repeat the above process until the exit node completes the encoding operation.

The data forwarding policy of the relay node is similar to the data forwarding policy of the source node. The relay node $R_{(i,j)}$ receives m packets sent by the relay node of the previous hop m-path. It selects the random coefficient and performs network encoding for m packets. Then, the encoded information, the ciphertext of the global encoding vector, and the local encoding row vector are forwarded along the m different paths to the next hop relay node. The specific steps for relay node data encoding are as follows:

Step 6. Random coefficient selection for the network encoding.

Randomly select m coding coefficient vectors of length d in a finite field F, ${C_{(i,j)}}^k$ ($i = 1,2, \ldots, m; j = 1,2, \ldots, m; k = 1,2, \ldots, m$), that forms a local coding matrix, denoted as $C_{(i,j)}$, i.e.,

$$C(i,j) = \begin{bmatrix} C^1_{(i,j)} \\ C^2_{(i,j)} \\ \vdots \\ C^m_{(i,j)} \end{bmatrix} (i = 1, 2, \ldots, m; j = 1, 2, \ldots, m). \quad (10)$$

Step 7. Relay node $R_{(i,j)}$ performs network encoding for the m packets received, $M_i^{(j)}$.

The local encoding matrix $C_{(i,j)}$ calculates a binary bit addition on $M_i^{(j)}$. That is, each encoding coefficient vector ${C_{(i,j)}}^k$ of $C_{(i,j)}$ performs an XOR operation on $M_i^{(j)}$, respectively, denoted as

$$\begin{aligned} M_i^{(j)} &= C_{(i,j)} \oplus M_i^{(j-1)} \\ &= \left(C^1_{(i,j)} \oplus M_i^{(j-1)}, C^2_{(i,j)} \oplus M_i^{(j-1)}, \ldots, C^m_{(i,j)} \oplus M_i^{(j-1)} \right) \\ &\quad (i = 1, 2, \ldots, m). \end{aligned} \quad (11)$$

Step 8. Calculating the global encoding vector ciphertext of the relay node $R_{(i,j)}$.

Since the global encoding vector is the ciphertext encrypted by the homomorphic encryption function in the received m-path packet, the relay node does not have the corresponding decryption key. The relay node could not decode the original packet to be directly recovered. At the same time, according to the homomorphic cryptographic function, linear changes could directly use the ciphertext of the global coding vector to generate new global coding vectors.

The global coding vector $V_i^{(j)}$ consists of the ith coding coefficient component, $C_{(i,j)}$; then, the global coding vector is as follows:

$$EV_i^{(j)} = \sum_{k=1}^{m} C^k_{(k,j-1)} EV_i^{(j-1)}. \quad (12)$$

Step 9. Forwarding the encoded message.

The relay node sends the encoded information $M_i^{(j)}$, the ciphertext of the global encoding vector $EV_i^{(j)}$, and the ith row vector of the local encoding matrix $C_{(i,j)}^{\ i}$ to the relay node in the ith path. In addition, the other row vectors of the local encoding matrix, $C_{(i,1)}^{\ k}$ $(k \neq i)$, are sent to the relay node along the j-th path, respectively.

Here, there are two points to be noted. For the entry node $R_{(i,1)}$, there is only one input link, $V_i^{(1)} = V_i^{(0)}$ and $EV_i^{(1)} = EV_i^{(0)}$. For the exit node, it broadcasts the encoded information $M_i^{(n)}$ and the ciphertext of the global encoding vector $EV_i^{(n)}$ and the local encoding matrix $C_{(i,j)}$ to the target vehicle node.

In the above encoding forwarding process, the encoded data slice $M_i^{(j)}$, the ciphertext of the global encoding vector $EV_i^{(j)}$, and the row vector of the local encoding matrix $C_{(i,j)}^{\ i}$ are forwarded along the unjoint m-paths, respectively. It prevents the relay node from recovering the original data.

Phase 4. The exit node broadcasts the encoding information, global encoding vectors, and local encoding matrix to the target node.

Phase 5. Without considering network errors, the target vehicle node D receives the network encoding information $M_i^{(n)}$, the global encoding vector ciphertext $EV_i^{(n)}$, and local encoding matrix $C_{(i,n)}$ from m-path. It uses the information to recover the information completing the data forwarding transmission process.

Assuming that the probability of transmission error is negligible, the target vehicle node D could receive the network encoding information $M_i^{(n)}$, the global encoding vector ciphertext $EV_i^{(n)}$, and the local encoding matrix $C_{(i,n)}$ from m-path. The decoding steps are as follows.

Step 10. The target node D uses the decryption key k_d to decrypt the ciphertext of the global encoding vector $EV_i^{(n)}$ to obtain the global encoding vector $V_i^{(n)}$, and the decryption operation is as follows:

$$\begin{aligned} V_i^{(n)} &= D\left(EV_i^{(n)}, k_d\right) \\ &= \left(\mathop{\oplus}_{k=0}^{n-1} C^i_{(1,k)}, \mathop{\oplus}_{k=0}^{n-1} C^i_{(1,k)}, \ldots, \mathop{\oplus}_{k=0}^{n-1} C^i_{(1,k)} \right). \end{aligned} \tag{13}$$

Step 11. The target vehicle node D restores the original slice information M_i according to the network encoding information $M_i^{(n)}$, the global encoding vector $V_i^{(n)}$, and the local encoding matrix $C_{(i,n)}$ as follows:

$$M_i = M_i^{(n)} \oplus \left(V_i^{(n)} \oplus C_{(i,n)}\right). \tag{14}$$

Step 12. The target vehicle node D recovers the original information M based on the original slice information M_i, denoted as follows:

$$M = \left(M_1, M_2, \ldots M_m\right). \tag{15}$$

Next, we will prove that the secure data delivery mechanism can be successfully decrypted after receiving encrypted data packets.

Assuming that the probability of transmission errors is negligible in a data forwarding network based on network encoding, the target vehicle node D could receive the network encoding information $M_i^{(n)}$, the ciphertext of global encoding vector $EV_i^{(n)}$, and the local encoding matrix $C_{(i,n)}$ from m-path.

Theorem 1. *Without considering network errors, the target vehicle node D receives the network encoded information $M_i^{(n)}$, the ciphertext of the global encoding vector $EV_i^{(n)}$, and the local encoding matrix $C_{(i,n)}$ from the m-path, using them to recover the information sent by the source node S correctly.*

Proof. Without considering network errors, the target vehicle node D receives the network encoding information $M_i^{(n)}$, ciphertext $EV_i^{(n)}$, and local encoding matrix $C_{(i,n)}$ from the m-path. The above information is obtained after n operations in the data forwarding network based on network encoding. First, analyze the calculation process of the above information.

The information slice M_i is encoded by n times to obtain $M_i^{(n)}$, and its calculation process is as follows:

$$\begin{aligned} M_i^{(n)} &= C_{(i,n)} \oplus M_i^{(j-1)} \\ &= C_{(i,n)} \oplus \left(C_{(i,n-1)} M_i^{(j-2)}\right) \\ &= C_{(i,n)} \oplus \left(C_{(i,n-1)} \oplus \left(C_{(i,n-2)} \oplus M_i^{(j-3)}\right)\right) \\ &= C_{(i,n)} \oplus \left(C_{(i,n-1)} \oplus \left(C_{(i,n-2)} \oplus \left(\cdots \left(C_{(i,0)} \oplus M_i\right)\right)\right)\right) \\ &= \mathop{\oplus}_{k=0}^{n} C_{(i,k)} \oplus M_i. \end{aligned} \tag{16}$$

The ciphertext of the global encoding vector $EV_i^{(n)}$ is obtained by calculating $EV_i^{(0)}$ by n times, and its calculation process is as follows:

$$\begin{aligned}
EV_i^{(n)} &= \sum_{k=1}^{n} C^k_{(k,n-1)} EV_i^{(n-1)} \\
&= \sum_{k=1}^{n} C^k_{(k,n-1)} \left(\sum_{k=1}^{n} C^k_{(k,n-2)} EV_i^{(n-2)} \right) \\
&= \sum_{k=1}^{n} C^k_{(k,n-1)} \left(\sum_{k=1}^{n} C^k_{(k,n-2)} \left(\sum_{k=1}^{n} \cdots \left(\sum_{k=1}^{n} C^k_{(k,1)} EV_i^{(1)} \right) \right) \right) \\
&= \sum_{k=1}^{n} C^k_{(k,n-1)} \left(\sum_{k=1}^{n} C^k_{(k,n-2)} \left(\sum_{k=1}^{n} \cdots \left(\sum_{k=1}^{n} C^k_{(k,1)} \left(EC^i_{(1,0)}, EC^i_{(2,0)}, \ldots, EC^i_{(m,0)} \right) \right) \right) \right) \\
&= \left(\bigoplus_{k=0}^{n-1} EC^i_{(1,k)}, \bigoplus_{k=0}^{n-1} EC^i_{(1,k)}, \ldots, \bigoplus_{k=0}^{n-1} EC^i_{(1,k)} \right).
\end{aligned} \tag{17}$$

The target vehicle node D uses the decryption key k_d to decrypt the global encoding vector ciphertext $EV_i^{(n)}$ and obtains the global encoding vector $V_i^{(n)}$. The result is as follows:

$$\begin{aligned}
V_i^{(n)} &= D\left(EV_i^{(n)}, k_d\right) \\
&= D\left(\left(\bigoplus_{k=0}^{n-1} EC^i_{(1,k)}, \bigoplus_{k=0}^{n-1} EC^i_{(1,k)}, \ldots, \bigoplus_{k=0}^{n-1} EC^i_{(1,k)} \right), k_d\right) \\
&= \left(\bigoplus_{k=0}^{n-1} C^i_{(1,k)}, \bigoplus_{k=0}^{n-1} C^i_{(1,k)}, \ldots, \bigoplus_{k=0}^{n-1} C^i_{(1,k)} \right).
\end{aligned} \tag{18}$$

According to $M_i^{(n)}$, $EV_i^{(n)}$, and the local encoding matrix $C_{(i,n)}$, the original information slice M_i can be recovered. The calculation method is as follows:

$$\begin{aligned}
M_i^{(n)} &= \bigoplus_{k=0}^{n} C_{(i,k)} \oplus M_i \\
&= \bigoplus_{k=0}^{n-1} C_{(i,k)} \oplus C_{(i,n)} \oplus M_i \\
&= V_i^{(n)} \oplus C_{(i,n)} \oplus M_i.
\end{aligned} \tag{19}$$

According to the above formula, M_i could be derived and calculated as follows:

$$M_i = M_i^{(n)} \oplus \left(V_i^{(n)} \oplus C_{(i,n)} \right). \tag{20}$$

Finally, the original information $M = (M_1, M_2, \ldots, M_n)$ could be recovered according to M_i. End. □

4. Security Analysis

This section proves the security of the model from confidentiality and anticollusion attack. The confidentiality is shown that the relay nodes except the target vehicle node could not obtain the original information. The anticollusion attack is shown that collusion attackers could not jointly recover the original data.

4.1. Confidentiality. In IoV-SDCM, the confidentiality of the message indicates that no node in the forwarding network could obtain the content of the sender message except for the destination node.

Theorem 2. *In addition to the target vehicle node D in IoV-SDCM, the relay nodes in the network could not recover some of the slicing information of the original information M sent by the source vehicle node S.*

Proof. According to the proof conclusions of reference [29], it was proved that the homomorphic encryption function encrypts the data homomorphically to form the ciphertext, and then, the obtained ciphertext calculation result is obtained by homomorphically decrypting the plaintext, which is the same as the result of directly calculating the plaintext data, but the plaintext data cannot be obtained. Based on the demonstration, the nodes in the IoV-SDCM forwarding network could not decrypt the part of the information of the ith hop global coding vector because the homomorphic encryption function encrypts the global encoding vector. The relay nodes except the target vehicle node D could not recover the original information M sent by the source vehicle node S. Therefore, there is no early decoding phenomenon, and the relay nodes could not decrypt part of the slicing information of the original information M. End.

In summary, IoV-SDCM ensures the confidentiality of the message. □

4.2. Anticollusion Attack. A collusion attack refers to the fact that multiple attackers collude with each other to decode the original information transmitted by the source vehicle node. The model proposed in this paper could prevent multiple leakers from conspiring to recover some information from the original information piece M_i.

Theorem 3. *In addition to the target vehicle node D in the model, there are multiple relay nodes in the anonymous forwarding network. The conspiratorial attackers could not obtain part of the original information M by leaking part of the information to recover the original data jointly.*

Proof. According to the evidence in the paper [30], when the global coding vector is exposed, multiple leak nodes could recover some of the information of the original information slice M_i through collusive attacks. The main reason is the leakage of the global coding vector $V_i^{(j)}$. By obtaining a partial global encoding vector $V_i^{(j)}$, an attacker could solve for some slices of the original information. The model proposed in this paper uses a homomorphic encryption function to encrypt the global encoding vector $V_i^{(j)}$. Only the target vehicle node D could decrypt the global encoding vector $V_i^{(j)}$, while other nodes could not decrypt the global encoding vector $V_i^{(j)}$. Therefore, the leakage of the global encoding vector $V_i^{(j)}$ is prevented, so that the attacker could not solve part of the slicing information. That is, the collusion attack of multiple leaked nodes could not be successful. End □

Corollary 1. *In addition to the destination node D in IoV-SDCM, if multiple relay nodes jointly recover the original information by disclosing some data, the conspiratorial attackers could not get the original data.*

Proof. According to Theorem 3, due to homomorphic encryption functions to encrypt the global encoding vector, the nodes in the forwarding network could not decrypt part of the i-hop global encoding vector. There is no early decoding phenomenon, and the conspiratorial attackers could not obtain part of the fragmented information of the original information M. Therefore, the intermediate nodes could not obtain the information of the original information M. End □

4.3. Privacy Protection. This section proves that the mode enabled privacy protection from packet analysis, traffic analysis, and packet tracing.

Packet analysis is when the attacker analyzes the packet to obtain information such as the identity and address of the sender or receiver. In the process of anonymous data forwarding, m packets are sent by the source vehicle node containing their pseudonyms and entry node identities; the m packets are sent by the relay node $R_{(i,j)}$ containing the relay node identities and successor node identities. Those nodes do not have the identity information of the actual source vehicle node and the target vehicle node. Only one packet in the m packets sent by the exit node $R_{(i,n)}$ contains the target vehicle node. The attacker could not distinguish which is the destination node. Therefore, in the process of anonymous data forwarding, an attacker could not obtain information such as the identity and address of the sender or receiver and nor could it determine the communication relationship between the sender and the receiver.

Traffic analysis is when an attacker determines the communication relationship by observing the traffic patterns of the network. In addition to the source vehicle node in the anonymous forwarding network, the input degree of other relay nodes $R_{(i,j)}$ is m, the output degree is also m, and the network traffic pattern is balanced. Therefore, the attacker could not determine the location of the target vehicle node by observing the network traffic pattern, but only the location of the source vehicle node could be found.

Packet tracing is when an attacker listens to a wireless channel near a node and determines the source vehicle node through hop-by-hop tracing. Assuming an attacker is listening to a wireless channel at a relay node in the communication cycle T, each relay node $R_{(i,j)}$ has m front-drive nodes. The attacker could move to a front-drive node $R_{(k,j\text{-}1)}$ of the listening node each time. If the attacker could move to the source vehicle node in the same communication cycle, you can locate the sender's location. Suppose the communication cycle ends and the attacker has not moved to the source vehicle node. In that case, the original anonymous forwarding network is abandoned, and the attacker could not correctly locate the sender's location. Therefore, the success of packet tracing is affected by the length of the communication cycle T and the forwarding path.

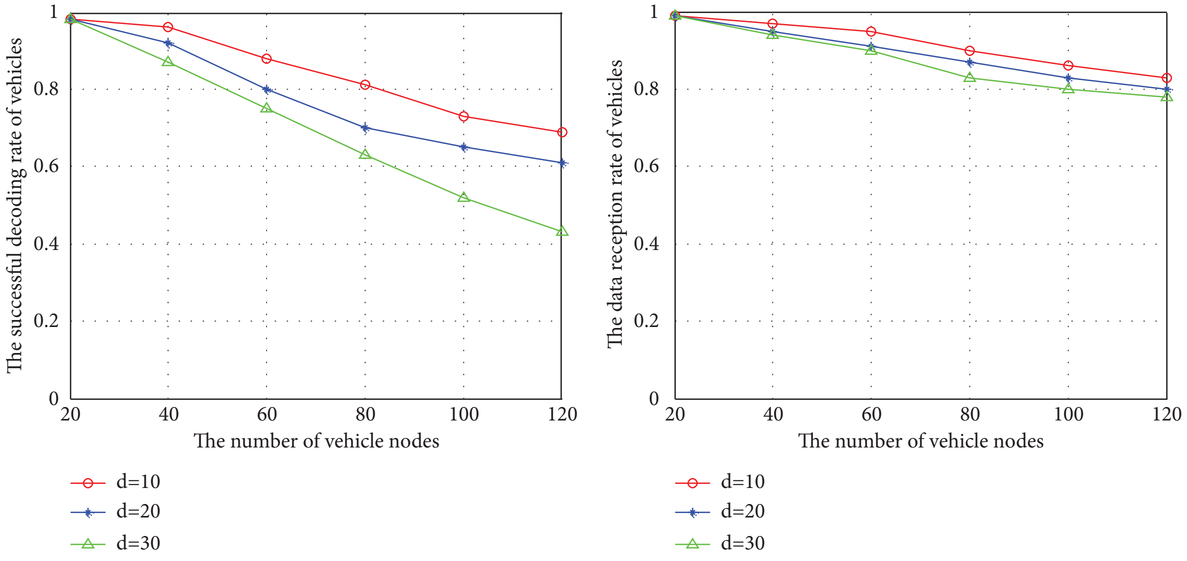
5. Performance Analysis

The IoV simulator of OMNeT++ is used to simulate the IoV environment. We set a one-way 4-lane road shape of 30000 m × 60 m. The vehicles only conduct V2V communication, the vehicle communication radius is 100~300 m, and the vehicle speed range is 50 km/h~100 km/h. The linear mobility mobile module is used to control the movement of nodes and the vehicle. The speed of the nodes is configured to follow a random distribution (14 mps, 28 mps). For the communication between vehicle nodes, the IEEE 802.11 b PHY/MAC protocol with a data rate of 11 Mbps is configured, and IoV-SDCM is added to the application layer module.

We give two definitions to analyze the data delivery performance of the model. One is the successful decoding rate of vehicle nodes, which is the number of nodes that can decode data divided by the number of all vehicle nodes in the simulation time. The other is the data receiving rate of vehicle nodes, which are the valid data packets received by all vehicle nodes divided by the total number of data packets in the simulation time.

In the experiment, set the slice size $d = 10$, 20, and 30, the size of each data packet after the slice is 1 MB, and the communication radius of the vehicle node is 200 m. Figure 4 shows that the experiment results in the successful decoding rate of vehicles decreases as the vehicle node amount increases. Compared with the successful decoding rate, the data reception rate decreases slowly. The network topology greatly affects the successful decoding rate of vehicles. When the vehicle is in a tight state, more data packets are obtained by V2V communication. However, the successful decoding rate of the vehicle is more demanding. It not only requires the vehicle to receive the data packets but also needs to be able to obtain enough packets to be decoded.

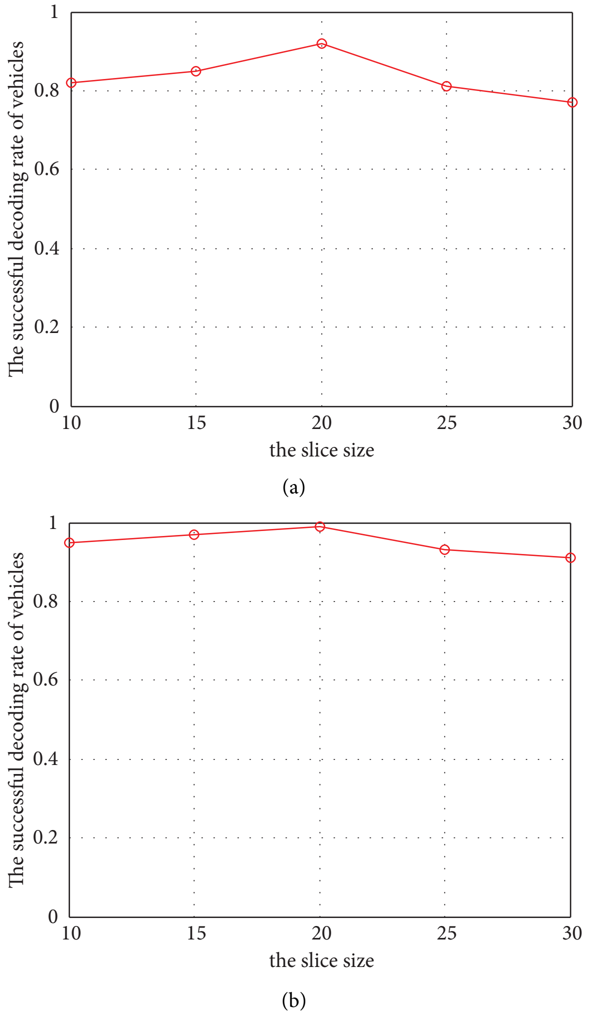
When the amount of data to be distributed is constant, the amount of data to be distributed is set to 30 MB, the communication radius of vehicle nodes is 200m, and the slice size is set to 10, 15, 20, 25, and 30, for a total of 100 vehicle nodes. Figure 5 shows that with the increase of the slice size d, the transmission time is reduced after the data volume of each data packet is reduced. It could ensure the successful receiving rate of vehicles, so the efficiency is increasing. At the same time, we can see that the change of the decoding rate is affected by the successful reception rate. The decoding rate is consistent with the change of the successful



(a)

(b)

Figure 4: The influence of the vehicle node amount on the delivery efficiency.



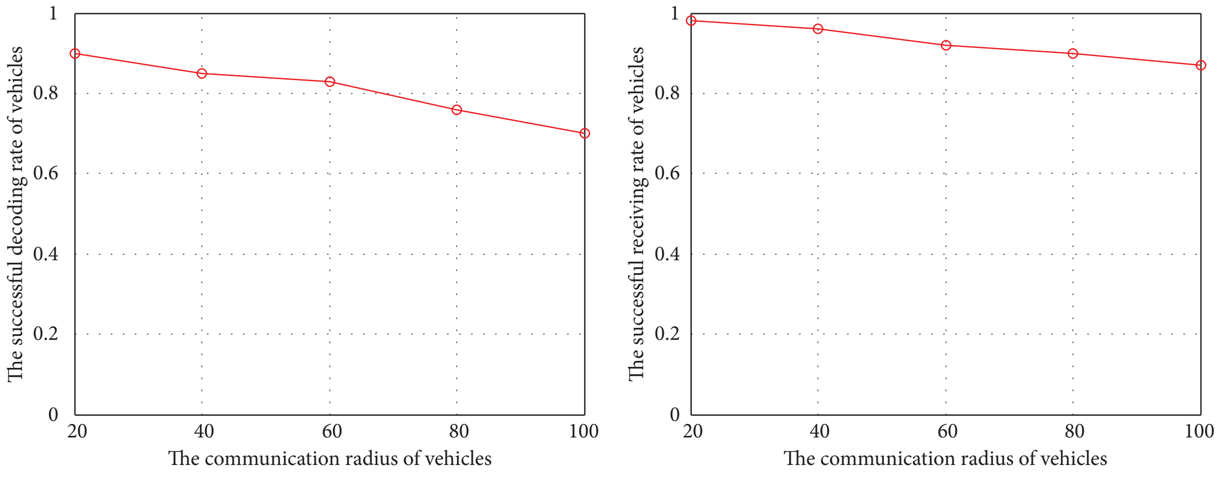
(a)

(b)

Figure 5: The influence of slice size on delivery efficiency.

```
Input: Source vehicle node S, target vehicle node D
Output: Anonymous forwarding network N
Process:
If (Period = $T$)
    $T$ = Null;//Using a tree structure to build an anonymous network and initialize the network.
    Send ($S$, $D$, mesg_rreq);//The source vehicle node $S$ sends a routing request message to the target vehicle node DRREQ, and record the path information from the source node to the target node.
    Path = Receive ($S$, $D$, mesg_path);//The target vehicle node $D$ sends an answer message containing the path information path to the source vehicle node $S$ and complete the construction of the forwarding network. See Initialization in 3.2 Section for detail.
    If (Path)//If there is a forwarding path between the source vehicle node and the target vehicle node.
        For (each Path [$i$], $i < n$)//For each node in the forwarding path, it is used as a group header to build an $m$-anonymous group.
            select $m - 1$ node satisfying ($Hop\_Group_i$ is in the radius of $Hop\_Group_{i\text{-}1}$ and $Hop\_Group_{i+1}$)
            $g_j = Hop\_Group_i[j]$;
        End For
    End If
    Path [$n$] = broadcast;//The exit node broadcasts messages to the target node
End If
End
```

Algorithm 1: An anonymous forwarding network construction.



(a)

(b)

Figure 6: The influence of communication range on delivery efficiency.

receiving rate, and the decoding rate is slightly lower than the receiving rate. However, the number of slices d can neither be too large nor too small. If it is too large, the reception will be incomplete and the decoding rate is low; if it is too small, the transmission times will be increased, and the system efficiency will be reduced and the decoding rate is also low.

With the expansion of the communication radius of vehicle nodes, the vehicle nodes can communicate with more nodes. In the simulation experiment, set the slice size $d = 10$, the size of each data slice is 3 MB, and there are 100 vehicle nodes. Figure 6 shows that the successful receiving rate and decoding rate of vehicles decreases, and the decrease is relatively stable if the vehicle communication radius increases. We can see that the decoding rate of data packets is consistent with the change of the successful reception rate, and the decoding rate is slightly lower than the receiving rate. Because when the communication radius becomes larger, the vehicle node can conduct V2V communication with more vehicles, so that the average request node increases, and the time that can communicate with other nodes is wasted, thus resulting in a decrease in efficiency.

6. Conclusion

This paper proposes IoV-SDCM, a secure data communication model in IoV. This model is based on the relay forwarding network and the secure data transmission mechanism. It constructs a dynamic communication network with the vehicle as the node. The data communication network reaches the target vehicle through the self-organizing relay collaboration policy with the security and privacy strategy. Theoretical proof proves that it ensures the confidentiality of data transmission with better antiattack capabilities, privacy protection capabilities, and simulation experiments verify that the model has the advantage of high and stable performance.

Data Availability

This paper adopts theoretical proof and simulation experiments to prove the correctness, security, and performance of the proposed model, so no experimental data are involved.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was funded by the Postdoctoral Research Fund of Guangzhou University.

References

[1] S. Chen, J. Hu, Y. Shi et al., "Vehicle-to-Everything (v2x) services supported by LTE-based systems and 5G," *IEEE Communications Standards Magazine*, vol. 1, no. 2, pp. 70–76, 2017.

[2] F. Wei, S. Zeadally, P. Vijayakumar, N. Kumar, and D. He, "An intelligent terminal based privacy-preserving multi-modal implicit authentication protocol for Internet of connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 3939–3951, 2021.

[3] R. E. Navas, F. Cuppens, N. Boulahia Cuppens, L. Toutain, and G. Z. Papadopoulos, "MTD, where art thou? A systematic review of moving target defense techniques for IoT," *IEEE Internet of Things Journal*, vol. 8, no. 10, pp. 7818–7832, 2021.

[4] T. A. Butt, R. Iqbal, K. Salah, M. Aloqaily, and Y. Jararweh, "Privacy management in social Internet of vehicles: review, challenges and blockchain based solutions," *IEEE Access*, vol. 7, Article ID 79694, 2019.

[5] J. Wan, X. Cao, K. Yao, D. Yang, E. Peng, and Y. Cao, "Data mining technology application in false text information recognition," *Mobile Information Systems*, vol. 2021, pp. 1–13, Article ID 4206424, 2021.

[6] X. Chen and G. Chu, "Data cooperative distribution mechanism of Internet of vehicles using D2D technology," *Advances in Multimedia*, vol. 2022, Article ID 9722915, 10 pages, 2022.

[7] M. Hosseini, R. Ghazizadeh, and H. Farhadi, "Game theory-based radio resource allocation in NOMA vehicular communication networks supported by UAV," *Physical Communication*, vol. 52, Article ID 101681, 2022.

[8] S. M. A. Kazmi, T. N. Dang, I. Yaqoob et al., "A novel contract theory-based incentive mechanism for cooperative task-offloading in electrical vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 8380–8395, 2022.

[9] T. Zeng, O. Semiariy, M. Chen, W. Saad, and M. Bennis, "Federated learning on the road autonomous controller design for connected and autonomous vehicles," in *Proceedings of the IEEE Transactions on Wireless Communications*, p. 1, Austin, TX, USA, December 2022.

[10] R. Ahlswede, C. Ning, S. Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[11] B. Zhang, Z. Liu, S. H. G. Chan, and G. Cheung, "Collaborative wireless freeview video streaming with network coding," *IEEE Transactions on Multimedia*, vol. 18, no. 3, pp. 521–536, 2016.

[12] T. Zhu, C. Li, Y. Tang, and Z. Luo, "On latency reductions in vehicle-to-vehicle networks by random linear network coding," *China Communications*, vol. 18, no. 6, pp. 24–38, 2021.

[13] H. Song, L. Liu, B. Shang, S. Pudlewski, and E. S. Bentley, "Enhanced flooding-based routing protocol for swarm UAV networks: random network coding meets clustering," in *Proceedings of the IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, pp. 1–10, Vancouver, BC, Canada, May 2021.

[14] A. Engelmann and A. Jukan, "Balancing the demands of reliability and security with linear network coding in optical networks," in *Proceedings of the 2016 IEEE International Conference on Communications*, pp. 1–7, Kuala Lumpur, Malaysia, May 2016.

[15] C. Cheng, J. Lee, T. Jiang, and T. Takagi, "Security analysis and improvements on two homomorphic authentication schemes for network coding," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 5, pp. 993–1002, 2016.

[16] D. Jiang, Y. Wang, Z. Lv, S. Qi, and S. Singh, "Big data analysis based network behavior insight of cellular networks for industry 4.0 applications," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1310–1320, 2020.

[17] R. Hussain, D. Kim, J. Son et al., "Secure and privacy-aware incentives-based witness service in social Internet of vehicles clouds," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2441–2448, 2018.

[18] T. Ho, M. Médard, J. Shi, M. Effros, and D. Karger, "On randomized network coding," in *Proceedings of the Annual Allerton Conference on Communication Control and Computing)*, Monticello, IL, USA, October 2003.

[19] T. Ho, M. Médard, R. Kötter, and R. K. David, "Toward a Random Operation of Networks," *IEEE Transactions on Information Theory - TIT*, vol. 50, 2004.

[20] K. Liu, J. K. Y. Ng, J. Wang, V. C. S. Lee, W. Wu, and S. H. Son, "Network-coding-assisted data dissemination via cooperative vehicle-to-vehicle/-infrastructure communications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 6, pp. 1509–1520, 2016.

[21] J. Kwon and H. Park, "Reliable data dissemination strategy based on systematic network coding in V2I networks," in *Proceedings of the 2019 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 744–746, Jeju, Korea, October 2019.

[22] C. Gao, Y. Li, Y. Zhao, and S. Chen, "A two-level game theory approach for joint relay selection and resource allocation in network coding assisted D2D communications," *IEEE Transactions on Mobile Computing*, vol. 16, no. 10, pp. 2697–2711, 2017.

[23] A. S. Khan and I. Chatzigeorgiou, "Opportunistic relaying and random linear network coding for secure and reliable communication," *IEEE Transactions on Wireless Communications*, vol. 17, no. 1, pp. 223–234, 2018.

[24] P. Xu, Z. Ding, and X. Dai, "Achievable secrecy rates for relay-eavesdropper channel based on the application of noisy network coding," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 7, pp. 1736–1751, 2018.

[25] J. Kang, R. Yu, X. Huang, and Y. Zhang, “Privacy-preserved pseudonym scheme for fog computing supported Internet of vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 8, pp. 2627–2637, 2018.

[26] F. Wang, Y. Xu, H. Zhang, Y. Zhang, and L. Zhu, “2FLIP: a two-factor lightweight privacy-preserving authentication scheme for vanet,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 2, pp. 896–911, 2016.

[27] U. Rajput, F. Abbas, and H. Oh, “A hierarchical privacy preserving pseudonymous authentication protocol for VANET,” *IEEE Access*, vol. 4, pp. 7770–7784, 2016.

[28] K. Rabieh, M. M. E. A. Mahmoud, and M. Younis, “Privacy-preserving route reporting schemes for traffic management systems,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 2703–2713, 2017.

[29] C. Gentry, *A Fully Homomorphic Encryption Scheme*, Stanford University, Stanford, CA, USA, 2009.

[30] C. Berge, *Graphs and Hypergraphs*, Elsevier, Amsterdam, Netherlands, 1973.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 8188977, 12 pages
https://doi.org/10.1155/2022/8188977

Research Article

A Blockchain-Enabled Trusted Protocol Based on Whole-Process User Behavior in 6G Network

Zhe Tu,[1,2] Huachun Zhou,[1,2] Kun Li,[1,2] Haoxiang Song,[1,2] and Yuzheng Yang[1,2]

[1]*School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China*
[2]*National Engineering Research Center of Advanced Network Technologies, Beijing 100044, China*

Correspondence should be addressed to Huachun Zhou; hchzhou@bjtu.edu.cn

Received 7 July 2022; Revised 18 August 2022; Accepted 15 September 2022; Published 11 October 2022

Academic Editor: Hui Xia

The access of massive users and devices in the 6G networks increases the risk of network attacks. Designing a trusted protocol to control user behavior can effectively improve the security capability of the network. However, most of the existing trusted protocols focus on unilateral user behavior and lack effective control over the whole process of user behavior. In this paper, we design a blockchain-enabled trusted protocol based on the whole-process user behavior. At first, we describe the Whole-Process User Behavior (WPUB) after the user accesses the network, and model the whole-process trusted control process. The proposed model establishes a trusted chain between user identity, access action, and communication traffic, and realizes the control of WPUB. Then, based on the proposed model, we design a whole-process trusted protocol with smart agents and smart contracts in combination with blockchain. Finally, we evaluate the designed protocol in the HyperLedger Fabric-based prototype system. Evaluations show that the proposed protocol can control the WPUB and reduce the risk of the network being attacked.

1. Introduction

The Sixth-Generation (6G) network realizes borderless connection under the global coverage, and enables the ubiquitous connectivity of massive users and devices by thoroughly integrating multiple heterogeneous networks, including satellite, air, ground, and sea networks [1–3]. The access of a large number of users and devices increases the potential risk of network attacks, bringing great challenges to network security [4–6]. The Trusted Protocol (TP) can effectively reduce the attacks launched by malicious users on the network by controlling and managing user behaviors, which is one of the important methods to improve network security [7–9]. How to construct a TP to detect malicious behaviors in 6G networks with massive connections is an urgent problem to be solved. However, traditional TPs (such as identity authentication, access control, and traffic detection) are mostly deployed in centralized networks and are difficult to be applied directly to 6G networks with dynamic changes in user behaviors and heterogeneous network structures. The 6G networks put forward new security requirements for TPs, which are mainly shown as follows.

(i) Behavior traceability. For the dynamically changing user behavior in 6G heterogeneous networks, TPs need to be able to memorize the user's historical behavior and make an accurate and dynamic control based on the user behavior [10, 11]. Besides, the data for TPs should be shared among trusted distributed nodes.

(ii) Privacy protection. User behavior data reflects the specific activities of users in the network [12, 13]. When analyzing user behavior, it should be ensured that user behavior data is not leaked and maliciously tampered with.

In recent years, as a key technology in the 6G network, blockchain has been widely used in various fields [14, 15]. The blockchain-based TPs can well meet the new security requirements of the 6G networks. On the one hand, storing user behaviors in the blockchain enables traceability of user historical behavior, making it possible to accurately control dynamically changing user behaviors. On the other hand, the decentralized and tamper-proof characteristics of blockchain ensure the security and reliability of the constructed blockchain-based TPs.

However, the existing blockchain-based TPs still have the following problems. Firstly, most of the existing methods manage user behavior under a single specific security requirement, and cannot comprehensively consider the whole-process user behavior after accessing the network. Secondly, the existing methods lack dynamic closed-loop feedback, and it is difficult to meet the needs of dynamic evaluation and closed-loop management. Therefore, it is urgent to construct a TP with dynamic closed-loop feedback that can comprehensively consider the whole-process user behavior.

In this paper, we design a Whole-Process User Behavior-based Blockchain-enabled Trusted Protocol (WPUB-BTP) that can control the whole-process user behavior after accessing the network. The proposed WPUB-BTP constructs a trusted control chain between user identity, access action, and communication traffic, and realizes the control of user behavior in the whole process. In addition, the protocol also builds dynamic closed-loop feedback based on user reputation, which realizes dynamic control of user behavior.

The contribution of this paper can be summarized as follows.

(i) We design the trusted control model of the whole-process user behavior, which can comprehensively consider identity authentication behavior, access control behavior, and communication traffic behavior.

(ii) We put forward a blockchain-enabled trusted protocol based on the proposed model to achieve dynamic control and closed-loop feedback on user behavior.

(iii) We evaluate the trusted protocol in a HyperLedger Fabric prototype system. The evaluation shows that the proposed protocol can control the whole-process user behavior after the user accesses the network, and reduces the risk of the network being attacked.

The remainder of this paper is organized as follows. In Section 2, we review the secure control methods for user behavior based on blockchain. In Section 3, we design the trusted control model of the whole-process user behavior consisting of identity authentication behavior, access control behavior, and communication traffic behavior. Based on the proposed model, we put forward the blockchain-enabled trusted protocol in Section 4. The prototype system and evaluation analysis of the WPUB-BTP are represented in Section 5. In the end, conclusions are drawn in Section 7.

2. Related Work

In this section, we review the related work on blockchain-based security control methods in three aspects: identity authentication, access control, and malicious traffic detection.

2.1. Blockchain-Based Authentication Method. Identity authentication prevents malicious users from accessing the network by identifying user identities. Recently, many researchers have designed many authentication methods based on blockchain technology to improve the security of the network.

In Vehicular Ad-hoc Networks (VANETs), Zheng et al. [16] proposed a blockchain-based authentication system, which can provide the trusted communication environment of the Vehicle to Vehicle (V2V) and Vehicle to Infrastructure (V2I). Similarly, Feng et al. [17] put forward a Blockchain-based Assisted Privacy-preserving authentication System (BAPS) for VANETs. The proposed system is efficient and scalable, and can efficiently achieve privacy-preserving authentication without any online registration center. In the Internet of drones, Feng et al. [18] presented a blockchain-based cross-domain authentication method to build an identity federation for collaborative domains. To ensure the privacy and security of the Intelligent Transportation Systems (ITS) networks, Qureshi et al. [19] proposed a Blockchain-based Privacy-Preserving Authentication model (BPPAU).

2.2. Blockchain-Based Access Control Method. The access control method can prevent malicious users from accessing network resources without authorization, and realize the management and control of user access behavior. With the development of blockchain, many blockchain-based access control methods have been proposed.

Tan et al. [20] suggested a blockchain-empowered general Green Smart Device (GSD) access control framework in the Green Internet of Things (GIoT). The proposed framework provides a fine-grained and extensible access control of GSDs and ensures the credibility and immutability of permission data and identity data during access. On the Internet of Things (IoT), Sun et al. [21] proposed a blockchain-based IoT access control system, which combines the permission blockchain, Attribute-Based Access Control (ABAC), and Identity-Based Signature (IBS) to achieve security, lightweight, and cross-domain access control. To provide decentralized Electrical Health Records (EHR) and service automation, a blockchain-based Internet of Medical Things (IoMT) architecture called Fortified-Chain is proposed by Egala et al. [22]. The proposed architecture can provide decentralized automation access control, security, and privacy. In the Industrial Internet of Things (IIoT), Feng et al. [23] put forward a novel access control framework based on blockchain, which consists of three types of chaincodes: PMC, ACC, and CEC. The proposed framework can achieve fast and reliable consensus based on historical behavior records stored in the ledger.

2.3. Blockchain-Based Traffic Detection Method. User traffic detection is another important way to improve network security. According to the way of traffic detection, it can be divided into methods-based statistical methods and methods based on machine learning methods [24]. In recent years, the development of blockchain has enabled more and more scholars to build detection models in blockchain networks based on existing traffic detection technologies.

In the Satellite Communication (SATCOM) systems, Cao et al. [25] proposed a blockchain-based access control and intrusion detection framework ACID, which can dynamically adjust the Access Control Rules (ACRs) and effectively detect attacks against smart contrasts. Similarly, Guo et al. [26] proposed a blockchain-based Distributed Collaborative Entrance Defense (DCED) framework to protect the satellite networks from malicious attacks. Experiment shows that the proposed framework can effectively protect the bandwidth resources of satellite Internet from DDoS attacks. Ramanan et al. [27] put forward a blockchain-based decentralized replay attack detection mechanism for large-scale power systems. The proposed mechanism can detect coordinated replay attacks with full privacy. To prevent IoT devices and other computing resources from DDoS attacks, Hayat et al. [28] proposed a Multilevel DDoS mitigation approach (ML-DDoS) based on blockchain. The results show that the proposed framework can accurately detect DDoS attacks in IoT, and has good performance in throughput, latency, and CPU utilization.

In Table 1, we summarize the relevant work of blockchain-based TPs and analyze whether they meet the security requirements of TPs in 6G networks. The above methods put forward the blockchain-based TPs to improve network security in different aspects. However, most methods only focus on one aspect of user behavior and lack control of the whole-process user behavior after accessing the network. In addition, for dynamically changing user behavior in the 6G network, those methods lack closed-loop feedback, and cannot adjust control strategies in real time according to user behaviors. Therefore, based on blockchain, we build a trusted protocol with dynamic closed-loop feedback to realize the whole-process behavior control of users, so as to meet the security requirements of TPs in the 6G networks.

3. Trusted Control Model

In this section, we first present the whole-process user behavior description. Then, we describe the trusted control model of the whole-process user behavior.

3.1. Whole-Process User Behavior Description. Before introducing the trusted control model, the Whole-Process User Behavior (WPUB) in the 6G network needs to be defined. According to the different behaviors initiated by users after accessing the network, the WPUB can be divided into three sub-behaviors: Identity Authentication Behavior (IAB), Access Control Behavior (ACB), and Communication Traffic Behavior (CTB), as shown below.

$$WPUB \triangleq \{IAB, ACB, CTB\}. \tag{1}$$

IAB is the description of authentication behavior when a user accesses the network. The IAB can be represented as a set consisting of Authentication Protocol (AP), Environment Attributes (EA), Identity Attributes (IA), Device Attributes (DA), etc., as shown in the following equation:

$$IAB \triangleq \{\mathrm{AP}, \mathrm{EA}, \mathrm{IA}, \mathrm{DA}, \ldots\}. \tag{2}$$

ACB describes the actions taken by the user to access the network resources, including Access Actions (AA), Resource Attributes (RA), User Privilege (UP), and Resource Privilege (RP). The ACB can be represented as

$$ACB \triangleq \{\mathrm{AA}, \mathrm{RA}, \mathrm{UP}, \mathrm{RP}, \ldots\}. \tag{3}$$

CTB reflects the behavior of the traffic generated by the user's interaction with other network entities after accessing the network. According to the granularity level of the traffic, CTB can be divided into Packet Behavior (PB), Flow Behavior (FB), Host Behavior (HB), Session Behavior (SB), etc., as shown in the following equation:

$$CTB \triangleq \{\mathrm{PB}, \mathrm{FB}, \mathrm{HB}, \mathrm{SB}, \ldots\}. \tag{4}$$

Therefore, according to the above equations (2-4), the WPUB can be expressed in detail as the follows:

$$WPUB \triangleq \begin{bmatrix} \mathrm{AP}, \mathrm{EA}, \mathrm{IA}, \mathrm{DA}, \ldots \\ \mathrm{AA}, \mathrm{RA}, \mathrm{UP}, \mathrm{RP}, \ldots \\ \mathrm{PB}, \mathrm{FB}, \mathrm{HB}, \mathrm{SB}, \ldots \end{bmatrix}. \tag{5}$$

3.2. Whole-Process Trusted Control Model. To realize the trusted control of the WPUB, a Whole-Process Trusted Control model (WPTC) deployed in the access gateway is proposed. According to the division of WPUB, WPTC can be divided into three different modules: Identity Authentication Module (IAM), Access Control Module (ACM), and Traffic Detection Module (TDM). The proposed three modules can control and manage the user's sub-behavior to ensure the trust of each process. Besides, to achieve closed-loop feedback and dynamic control between three different control processes, a Dynamic Control Mechanism (DCM) based on the user's reputation is also proposed. The DCM constructs a dynamic control between user sub-behaviors in different modules and realizes the trusted control of whole-process behavior. The WPTC is shown in Figure 1.

3.2.1. Identity Authentication Module. The IAM authenticates the identity of users to ensure the trusted user identity, which is the first security protection barrier in the WPTC framework. To better model the IAM and reflect the control process of the module on IAB, we represent the Identity Authentication Result (IAR) as the mapping relationship of IAB, as shown in the following equation:

$$\begin{bmatrix} IAR_1^t \\ \cdots \\ IAR_i^t \\ \cdots \\ IAR_n^t \end{bmatrix} \triangleq f\left(\begin{bmatrix} IAB_1^t \\ \cdots \\ IAB_i^t \\ \cdots \\ IAB_n^t \end{bmatrix}\right). \tag{6}$$

Table 1: Analysis of related work.

Ref.	Year	Security requirement of trusted protocol in 6G networks					
		Trusted user identity	Trusted access actions	Trusted communication traffic	Closed-loop feedback	Privacy protection	Behavior traceability
[16]	2019	✓	×	×	×	✓	✓
[17]	2019	✓	×	×	×	✓	✓
[18]	2021	✓	×	×	×	✓	✓
[19]	2022	✓	✓	×	×	✓	✓
[20]	2021	×	✓	×	×	✓	✓
[21]	2021	×	✓	×	×	✓	✓
[22]	2021	✓	✓	×	×	✓	✓
[23]	2021	×	✓	×	×	✓	✓
[25]	2021	×	✓	✓	×	✓	✓
[26]	2022	×	×	✓	×	✓	✓
[27]	2021	×	×	✓	×	✓	✓
[28]	2022	×	×	✓	×	✓	✓
Ours	2022	✓	✓	✓	✓	✓	✓

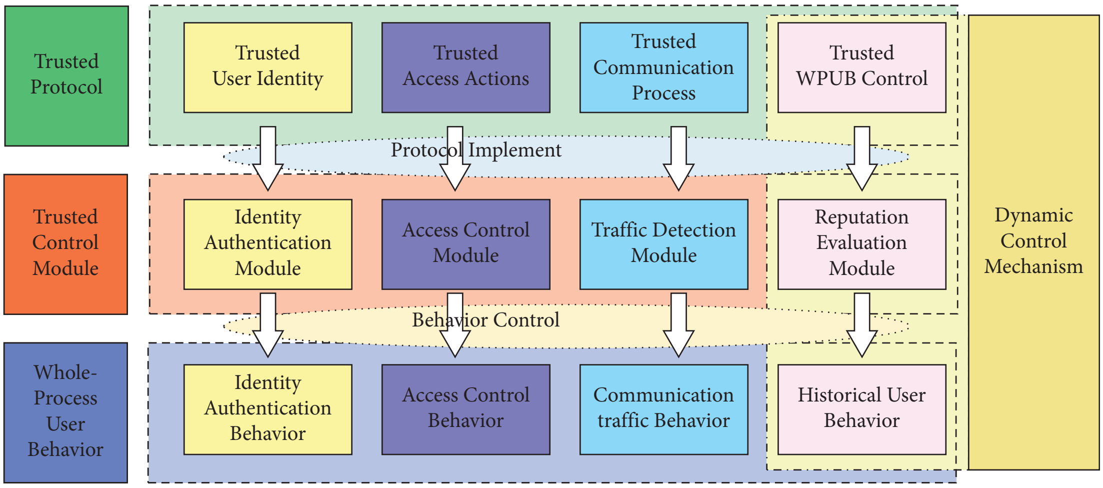


Figure 1: The framework of the whole-process trusted control model.

It is assumed that n users are accessing the network through the access gateway at time t. In (6), f is the trusted authentication protocol reflecting the relationship between IAR and IAB. IAB_i^t and IAR_i^t represent the IAB and IAR of user u_i at time t, respectively. If the identity of user u_i is trusted, the IAR_i^t is set to 1. Otherwise, IAR_i^t is set to 0. $1 \leq i \leq n$.

$$\begin{bmatrix} ACR_1^t \\ \cdots \\ ACR_i^t \\ \cdots \\ ACR_n^t \end{bmatrix} \triangleq g\left(\begin{bmatrix} ACB_1^t \\ \cdots \\ ACB_i^t \\ \cdots \\ ACB_n^t \end{bmatrix}, \begin{bmatrix} IAR_1^t \\ \cdots \\ IAR_i^t \\ \cdots \\ IAR_n^t \end{bmatrix} \right). \tag{7}$$

3.2.2. Access Control Module. The ACM is the key module to ensure the trust of access actions, which verifies whether the user can be authorized to access the Network Resources (NR) according to the access policy. The user needs to be authenticated before performing access control. A user with a trusted identity can access the network resources only after obtaining the legitimate access authorization. The ACM can be modeled as shown in (7). g() is the trusted access control protocol. ACB_i^t and ACR_i^t represent the ACB and the Access Control Result (ACR) of user u_i at time t, respectively. If the access action of u_i is authorized, the access control result is 1. Otherwise, ACR_i^t is 0.

3.2.3. Traffic Detection Module. As an important component in WPTC, TDM detects the traffic in the network in real time and realizes the timely detection and blocking of malicious CTB. The TDM module provides a guarantee for the trust of the communication traffic. In the proposed WPTC, the user can only send traffic to the NR after obtaining access authorization. Therefore, we define the trusted traffic detection protocol in TDM as the mapping relationship between CTB, ACR, and Traffic Detection Results (TDR). $h()$ is the trusted traffic detection protocol. CTB_i^t and TDR_i^t, respectively, represent the CTB and the ACR of user u_i at time t. If the traffic initiated by u_i is detected as normal, then TDR_i^t is 1; if the CTB_i^t is detected as malicious traffic, TDR_i^t is 0.

$$\begin{bmatrix} TDR_1^t \\ \cdots \\ TDR_i^t \\ \cdots \\ TDR_n^t \end{bmatrix} \triangleq h\left(\begin{bmatrix} CTB_1^t \\ \cdots \\ CTB_i^t \\ \cdots \\ CTB_n^t \end{bmatrix}, \begin{bmatrix} ACR_1^t \\ \cdots \\ ACR_i^t \\ \cdots \\ ACR_n^t \end{bmatrix}\right). \tag{8}$$

3.2.4. Dynamic Control Mechanism. The above three modules control user sub-behaviors from three aspects: user identity, access action, and communication traffic. By constructing a trusted control chain of the "user identity-access action-communication traffic," WPTC realizes the security control of user behavior in the whole process. In order to improve the security capability of closed-loop feedback and dynamic control, we introduce the DCM in WPTC.

DCM is the core control mechanism of WPTC, which can dynamically control the user's behavior by evaluating the reputation of the user. In DCM, the user's reputation is calculated by the Reputation Evaluation Module (REM), and the reputation is consisting of two kinds of subreputations: Sub-behavior Reputation (SR) and Global Reputation (GR). The SR is calculated by the historical behavior of each sub-behavior. Based on the division of the WPUB, the SR of user u_i at time t can be subdivided into user identity reputation UIR_i^t, access action reputation AAR_i^t, and communication traffic reputation CTR_i^t. The UIR_i^t, AAR_i^t, and CTR_i^t can be calculated by (9–11), respectively.

$$\begin{aligned} UIR_i^t &= \varphi_1\left(IAB_i^T\right) \\ &= \varphi_1\left(IAB_i^{t1}, \ldots, IAB_i^{tm}\right), \end{aligned} \tag{9}$$

$$\begin{aligned} AAR_i^t &= \varphi_2\left(ACB_i^T\right) \\ &= \varphi_2\left(ACB_i^{t1}, \ldots, ACB_i^{tm}\right), \end{aligned} \tag{10}$$

$$\begin{aligned} CTR_i^t &= \varphi_3\left(CTB_i^T\right) \\ &= \varphi_3\left(CTB_i^{t1}, \ldots, CTB_i^{tm}\right). \end{aligned} \tag{11}$$

In (9)–(11), IAB_i^T, ACB_i^T, and CTB_i^T represent the historical sub-behaviors of IAB, ACB, and CTB in the time period T before time t, respectively. IAB_i^{t1} is the first historical sub-behavior IAB of u_i in the time period T. Likewise, the historical sub-behavior in the time period of ACB and CTB can be represented similarly to the IAB. φ_1, φ_2, and φ_3 are the reputation evaluation functions of IAB, ACB, and CTB, respectively.

The global reputation GR_i^t of user u_i can be calculated by the above three sub-behavior reputations, as shown in (12). θ is the global reputation calculation function.

$$\mathrm{GR}_i^t = \theta\left(UIR_i^t, AAR_i^t, CTR_i^t\right). \tag{12}$$

When the user behavior is untrusted, based on proposed SR ($UIR_i^t, AAR_i^t, CTR_i^t$) and GR ($\mathrm{GR}_i^t$), we put forward the DCM in the above three models. The DCM can be divided into the following three stages.

In the identity authentication stage, the Dynamic Control Result (DCR) generated by DCM can be modeled as (13). When the identity of user u_i is untrusted ($IAR_i^t = 0$), the DCM can formulate different DCRs according to the different UIR_i^t. μ_1 is the security control judgment function of DCM in the IAM, and $DC\ R_i^t$ is the DCR of user u_i at time t. If UIR_i^t is greater than the threshold value ω, the DCR_i^t of user u_i is set to "re-authenticate." If $UIR_i^t < \omega$, the DCR_i^t is set to "access blocking," and the user is not allowed to access the network.

$$DCR_i^t = \mu_1\left(UIR_i^t | IAR_i^t = 0\right). \tag{13}$$

In the access control stage, the dynamic control process can be represented as (14). The DCM in the ACM ensures that different control policies are implemented based on different UIR_i^t and AAR_i^t when user's access behaviors are abnormal ($ACR_i^t = 0$). μ_2 is the security control judgment function of DCM in the ACM. If the access reputation value AR_i^t of user u_i is less than the threshold value λ_1, DCR_i^t is "access blocking," which means the access behavior of the user is blocked. If $\lambda_1 \leq AR_i^t < \lambda_2$, the user needs to be re-authenticated; If $\mathrm{AR}_i^t \geq \lambda_2$, the DCR_i^t is "re-access control," and the user needs to perform access control again. The AR_i^t can be calculated as follows. $\mathrm{AR}_i^t = \psi(UIR_i^t, AAR_i^t)$. ψ is the evaluation function of the access behavior.

$$DCR_i^t = \mu_2\left(AR_i^t | ACR_i^t = 0\right). \tag{14}$$

In the traffic detection stage, DCM can be modeled as (15). When a user initiates abnormal traffic to the network ($CTB_i^t = 0$), DCM formulates different security control schemes based on the user's global reputation GR_i^t to improve the security capability of the network. μ_3 indicates the security control judgment function of the DCM in the ACM. When the user traffic is detected as malicious traffic, the communication traffic is blocked. If the global reputation GR_i^t is less than ρ_1, the user is recorded on the blacklist and is not allowed to access the network for a period of time. If $\rho_1 \leq \mathrm{GR}_i^t < \rho_2$, the DCR_i^t is "re-authenticate"; If $\mathrm{GR}_i^t \geq \rho_2$, the user should be "re-access control." ρ_1 and ρ_2 are the threshold constants of global reputation in the traffic detection stage.

$$DCR_i^t = \mu_3\left(GR_i^t | CTB_i^t = 0\right). \tag{15}$$

In (13–15), the DCR_i^t is one of the elements in the set of Dynamic Control Policies (DCP). $DCR_i^t \in DCP$. DCP can be given as follows:

$$DCP = \{dcp_1, \ldots, dcp_n\}. \tag{16}$$

In (16), dcp_n is the nth subcontrol policy in the DCP set. In the DCM, the subcontrol policy dcp_n can be set as "re-authentication," "re-access control," "access blocking," "traffic blocking," and so on according to the specific network scenario.

4. Blockchain-Enabled Trusted Protocol Based on WPUB

In this section, based on the proposed trusted control model, we design the Blockchain-enabled Trusted Protocol (WPUB-

BTP) including trusted user identity protocol, trusted access action protocol, and trusted communication traffic protocol.

In WPUB-BTP, the functions of the modules in the trusted control model are deployed in the access gateway and blockchain network in the form of Smart Agents (SA) and Smart Contracts (SC). The SA is mainly responsible for interacting with UEs, processing and forwarding the user requests, while the SC stores the user behaviors and generate trusted management policies in the blockchain.

The division of modules in the trusted control model can be shown as follows. The functions of the IAM are performed by the Identity Authentication Agent (IAA) and Identity Authentication Smart Contract (IASC), and the ACM is deployed as the Access Control Agent (ACA) and Access Control Smart Contract (ACSC). In addition, the TDM is deployed in WPUB-BTP as a Traffic Detection Agent (TDA) and Traffic Detection Smart Contract (TDSC). The Reputation Evaluation Smart Contract (RESC) in the blockchain network is deployed to perform the functions of the proposed REM. Besides, the user in the WPUB-BTP is represented as UE, and the network resources in the servers are abbreviated as NR.

In the following subsections, we will describe the three subprotocols in WPUB-BTP for security control of user subbehaviors. The blockchain-enabled trusted protocol is shown in Figure 2.

4.1. Trusted User Identity Protocol. In the trusted user identity protocol, the IAA is used to forward and process the identity authentication requests of users, while the IASC stores the authentication credentials and generates the user authentication vector.

The trusted user identity protocol can be described as the following steps.

STEP 1: UE sends the authentication request to IAA;

STEP 2: IAA invokes the interface of IASC to generate authentication vector and authenticate user identity. If the user identity is authenticated successfully, go to STEP 4. Otherwise, go to STEP 3.

STEP 3: If the user identity is untrusted, IAA needs to query the User Identity Reputation (UIR) of the user, and generates the DCR according to the UIR;

STEP 4: Meanwhile, the IAA invokes IASC interfaces to record identity authentication behaviors.

STEP 5: RESC updates the user identity reputation based on the recorded IAB;

STEP 6: Finally, IAA returns the IAR or the DCR to UE.

4.2. Trusted Access Action Protocol. The trusted access action protocol in the WPUB-BTP is used to evaluate user access control behavior. In the trusted access action protocol, there are two components, ACA and ACSC, which perform the access control function. The ACA is used to forward the access control requests initiated by users, while the ACSC generates the access policy and stores the user access control behavior.

The trusted access action protocol consists of the following seven steps.

STEP 1: UE sends the access control request to the ACA.

STEP 2: After receiving the access control request, the ACA looks up the identity authentication result of the UE to verify whether the user identity is legal; If the user is illegal, the ACR is set to 0, and the next step is STEP 5. Otherwise, go to STEP 3.

STEP 3: If the identity of the user is trusted, the ACSC generates the access control policy for the UE. If the user access action is unauthorized, go to STEP 4. Otherwise, go to STEP 5.

STEP 4: ACA queries the user's Access Action Reputation (AAR), and generates the DCR based on the obtained AAR.

STEP 5: At the same time, the ACA invokes ACSC interfaces to record access control behaviors.

STEP 6: RESC updates the access action reputation based on the recorded ACB.

STEP 7: In the end, ACA returns the access control result or the dynamic control result to UE.

4.3. Trusted Communication Traffic Protocol. In the trusted communication traffic protocol, the TDA in the access gateway is the component that mainly performs the function of traffic detection. In TDA, different types of detection submodules can be deployed to detect the user traffic passing through the gateway in real time. The TDSC in the protocol periodically stores the communication traffic behavior of users.

The trusted communication traffic protocol is used to control the communication traffic behavior of users, which includes the following steps.

STEP 1: UE sends the communication traffic through the access gateway to the NR.

STEP 2: The TDA in the access gateway needs to ask the ACSC contract whether the user has permission to access NR when the user's traffic arrives for the first time.

STEP 3: If the UE is an authorized access user, the user is allowed to send traffic to network resources. At the same time, the TDA continuously detects the traffic between UE and NR in real time.

STEP 4: If the traffic initiated by the user is detected abnormal, the communication traffic needs to be blocked at the first time. Then, the TDA calls the interface of RESC to obtain the user's Communication Traffic Reputation (CTR), and generates the DCR based on the obtained CTR;

STEP 5: Meanwhile, the TDA periodically records the CTB in the TDSC contract based on the traffic detection results.

STEP 6: And the RESC updates the communication traffic reputation based on the recorded CTB.

STEP 7: At last, the TDA returns the dynamic control result to UE.

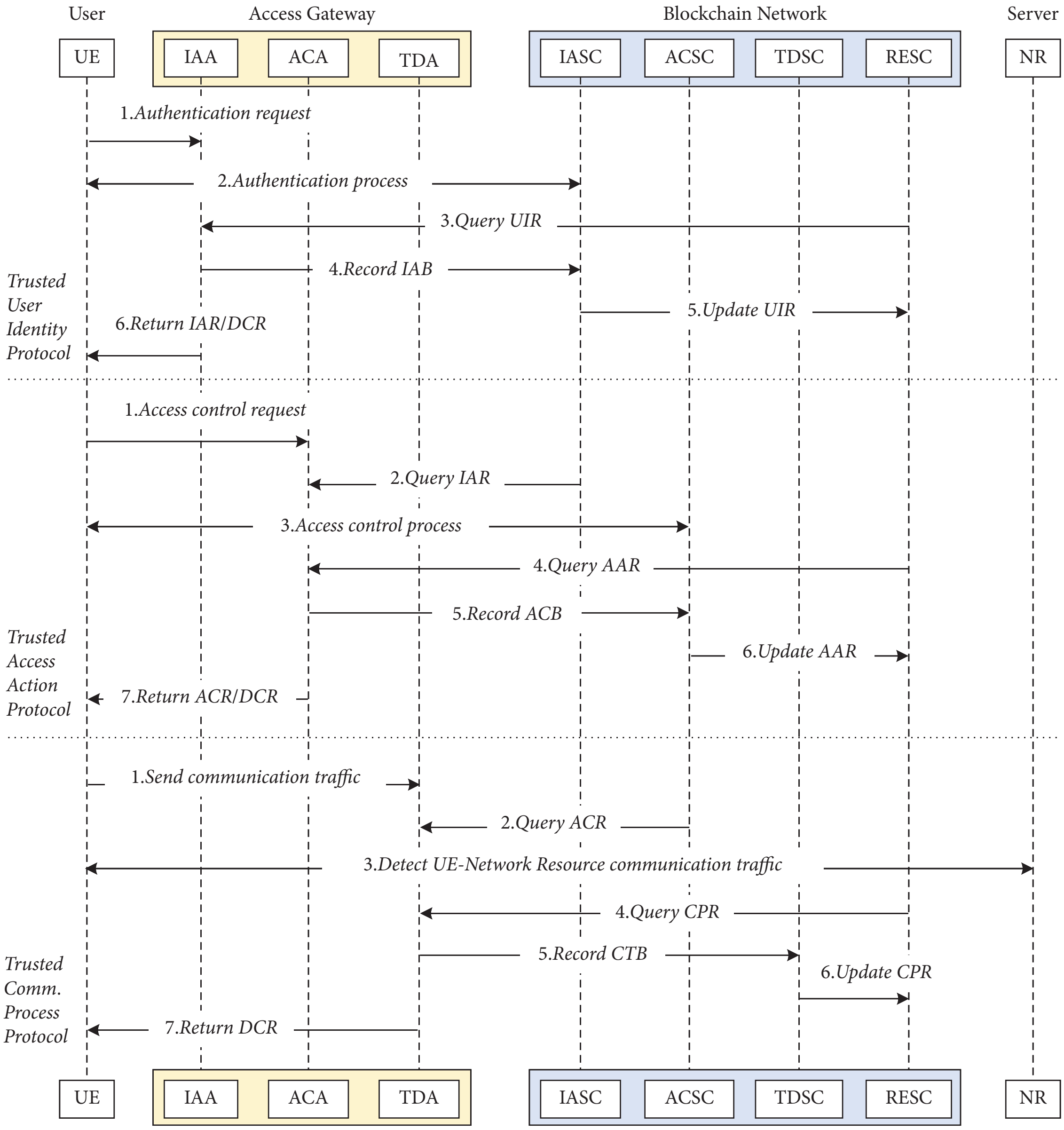


Figure 2: The blockchain-enabled trusted protocol.

5. Evaluation

In this section, we first introduce the prototype system based on the proposed WPUB-BCP protocol. Then, we evaluate the WPUB-BCP protocol in the HyperLedger Fabric prototype system.

5.1. Prototype System. As shown in Figure 3, based on the proposed WPUB-BTP protocol, a prototype system is deployed for evaluation. We deploy a server cluster based on VMware vSphere [29] virtualization platform. The server cluster consists of 12 servers, each configured with a 40G disk, 16G memory, and an 8-core processor. In the server cluster, 12 servers can be divided into satellite networks domain, cellular networks domain, and wireless local area networks domain depending on the application scenario. And each domain contains one UE and three access gateways.

Compared with other blockchain platforms such as Ethereum (https://ethereum.org/), HyperLedger Fabric (https://github.com/hyperledger/fabric/) has the advantages of high modularity and scalability, and has been widely and maturely applied in various commercial scenarios. Therefore, in this article, we build the WPUB-BTP protocol prototype system based on Fabric. In the prototype system, the blockchain network is constructed on the nine access gateways.

In the prototype system, the HyperLedger Fabric blockchain network is divided into three organizations (3 Org), and each organization consists of one certificate authority (1 CA), three peer nodes (3 peers), and one

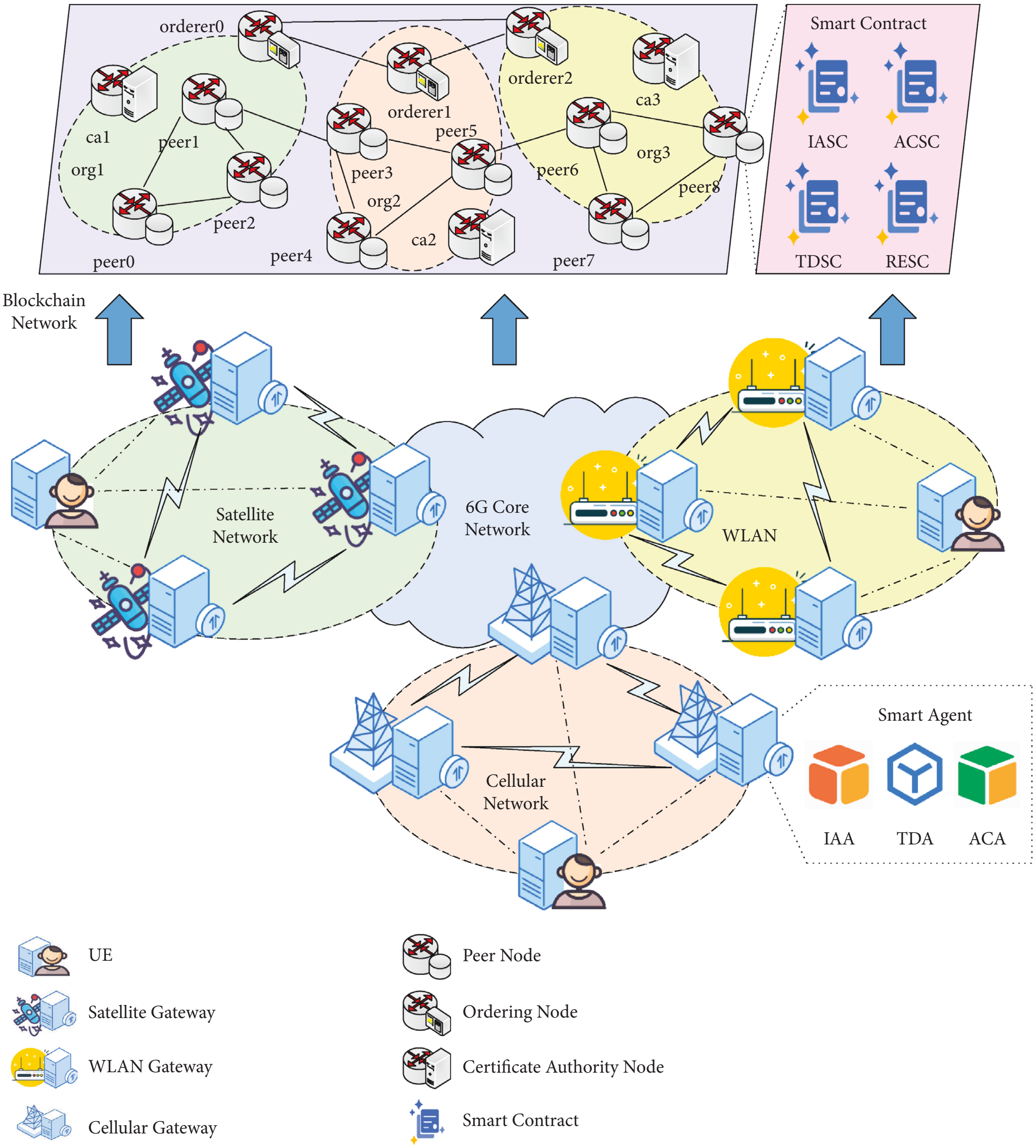


Figure 3: The prototype system of WPUB-BTP protocol.

ordering node (1 orderer). The access gateways initiate the transactions to the blockchain network through the SDK interface (fabric-py-sdk (https://github.com/hyperledger/fabric-sdk-py/)) for data storage, update, and query operations. Three smart agents (IAA, ACA, and TDA) written in Python (https://docs.python.org/3.9/) are deployed at each access gateways, performing identity authentication, access control, and traffic detection functions. In addition, we design four smart contracts (IASC, ACSC, TDSC, and RESC) based on the go-lang (https://github.com/golang/go/) language and deploy them in the blockchain network in the form of chaincodes. IASC and ACSC are used to control user authentication behavior and access control behavior, respectively. TDSC is used to detect the traffic behavior sent by users, while RESC evaluates the reputation based on user authentication, access control, and traffic behavior to realize dynamic closed-loop control of user behavior.

To evaluate the performance of the proposed WPUB-BCP protocol, we deploy the specific control methods in each module (SA and SC). In our previous work [30], an authentication method based on EAP-MD5 is proposed for fast authenticate. Therefore, in the IAM module, we use the same authentication method to represent the trusted authentication protocol f, so as to ensure the trusted user identity. Besides, an access control method based on the

Attribute-Based Access Control (ABAC) model [31] is deployed in the ACM module to represent the trusted access control protocol g. In the TDM, we deploy the same traffic detection method based on the Deep Deterministic Policy Gradient (DDPG) algorithm as in [32] to represent the trusted communication traffic protocol h. In addition, the Beta Reputation System (BRS) [33] can give a comprehensively evaluation of users' positive and negative behaviors. Therefore, in this paper, we deploy the BRS in REM module to evaluate the reputation of user's sub-behavior $(UIR_i^t, AAR_i^t, CTR_i^t)$ and to provide the feedback for dynamic control. φ_1, φ_2, and φ_3 are the reputation value calculation formulas of beta reputation system. Specifically, the global reputation and the access reputation can be calculated as follows: $\mathrm{GR}_i^t = 1/3 * (UIR_i^t + AAR_i^t + CPR_i^t), \mathrm{AR}_i^t = 1/2 * (UIR_i^t + AAR_i^t)$. In addition, the threshold constants in the DCM are set as follows: $\omega = 0.5, \lambda_1 = 0.35, \lambda_2 = 0.65, \rho_1 = 0.4, \rho_2 = 0.7$. μ_1, μ_2, and μ_3 are set as described in Section 3.2.

5.2. Performance Evaluation. In this subsection, we first evaluate the performance of the three proposed trusted protocols: trusted user identity protocol, trusted access action protocol, and trusted communication traffic protocol. Subsequently, we functionally evaluated the designed dynamic control mechanism.

5.2.1. Trusted User Identity Protocol. Figure 4 shows the evaluation result of the trusted user identity protocol. We evaluate the control results of the trusted user identity protocol under 100, 500, 1000, 2000, 5000, and 10000 authentication requests, and the proportion of illegal users is 20%, 40%, 60%, and 80%, respectively. As can be seen from Figure 4, the proposed trusted user identity protocol can achieve accurate authentication of a large number of users. In addition, the proposed protocol can prevent illegal users from accessing the network, which improves the security of the network.

5.2.2. Trusted Access Action Protocol. Subsequently, we evaluate the trusted access action protocol with 100, 200, 500, and 1000 access control requests per second in, as shown in Figure 5. In the evaluation, it is assumed that 20% of the requests are sent by unauthenticated UEs and 80% by the trusted identity UEs. In addition, it is assumed that 60% of users with trusted identities can obtain access policies. As can be seen from Figure 5, the proposed trusted access action protocol can evaluate user access control behaviors and successfully generate the corresponding access policies. Furthermore, the evaluation results show that users without trusted identities cannot get access authorization, which ensures the security and credibility of the network from both user identity and access action.

5.2.3. Trusted Communication Traffic Protocol. In Figure 6, the management and control process of user traffic behavior by the proposed trusted communication traffic protocol is shown. We simulated the traffic sent by two types of authorized users, namely normal user traffic and abnormal user traffic. Within 0–200 s, the normal users continuously send normal traffic to the network resource, while the abnormal uses periodically launch attack traffic. Both the normal traffic and the abnormal traffic are generated according to traffic dataset collected in [20]. The traffic detection module is deployed in the access gateway at 50 s. As shown in the figure, the traffic detection module can distinguish the normal traffic and abnormal traffic according to the traffic characteristics. And the trusted communication traffic protocol can generate the real-time control policies to block the malicious traffic according to the detection results.

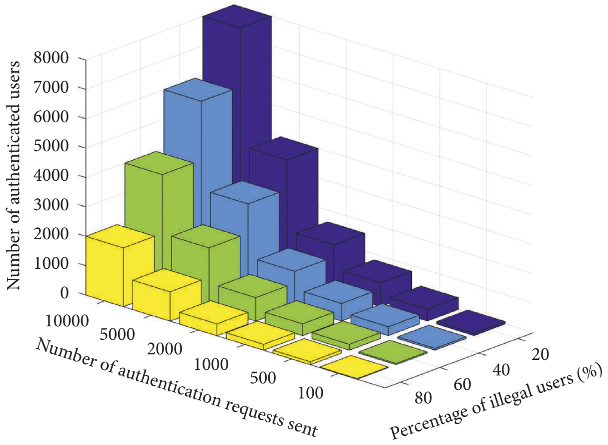


FIGURE 4: The evaluation of the trusted user identity protocol.

5.2.4. Dynamic Control Mechanism. In this subsection, we evaluate the continuous dynamic control results of the proposed dynamic control mechanism on user behavior when the user accesses the network and performs identity authentication, access control, and traffic detection in sequence.

As shown in Table 2, we simulate the user behavior of 200 users accessing the network. At the beginning of 200 users accessing the network, we set 50% of users to send correct authentication requests, 25% of high-reputation users (reputation greater than 0.5) to send incorrect authentication requests, and 25% of low-reputation users (low reputation greater than 0.5) to send a bad authentication request. The 100 users with trusted identities who send correct authentication requests need to perform access control when accessing network resources. Similarly, we set the following settings for users who send access control requests, among which 50% of users have successful access control, and 50% of users have failed access control; among the users whose access control fails, we set 50% of the users whose reputation is higher than 0.65, 30% of users have a reputation between 0.35 and 0.65, and 20% of users have a reputation below 0.35. Finally, among the 50 authorized users, we set 25 users send normal traffic, and the rest send abnormal traffic. In order to display the dynamic control results in the traffic detection stage, we divided the users sending abnormal traffic into three groups as follows: good reputation (reputation is greater than 0.7), moderate

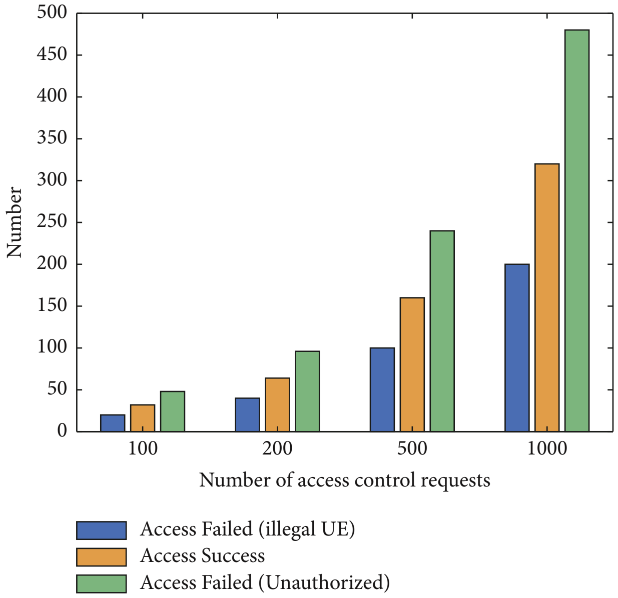


Figure 5: The evaluation of the trusted access action protocol.

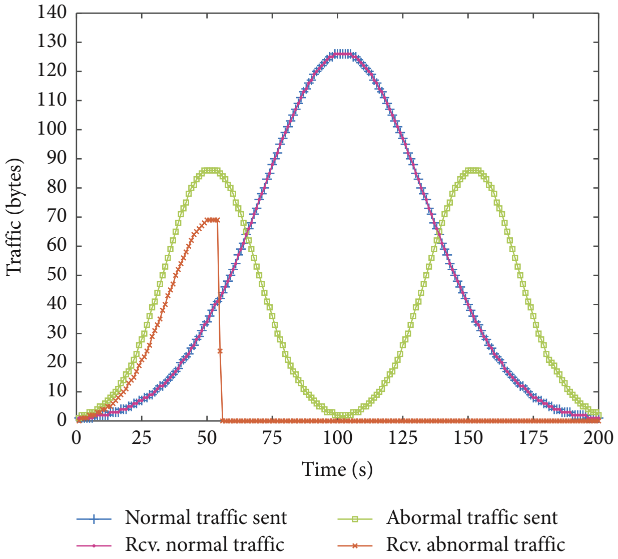


Figure 6: The evaluation of the trusted communication traffic protocol.

reputation (reputation is between 0.4 and 0.7), and low reputation (reputation is lower than 0.4). The three groups have 25, 15 and 10, users respectively.

Figure 7 shows the dynamic control results of the whole-process user behavior in three continuous stages. 0–200 s is the user identity authentication stage; 200–300 s is the user access control stage; and 300–350 s is the user traffic detection stage. It should be noted that, in order to visually display the results of dynamic control mechanism, Figure 7 only shows the number of users who successfully authenticated for the first time and access control for the first time, but does not show the number of users who successfully re-authenticated and re-access control.

In the identity authentication stage, we simulated a total of 200 users sending identity authentication requests to IAM. As can be seen from Figure 7, the designed IAM can accurately control user authentication behavior, and can generate different dynamic control results according to different reputation values of users.

Only users who are successfully authenticated in the identity authentication stage can perform access control. Therefore, in the access control stage, it can be seen from Figure 7 that the number of re-authentication ("re-auth"), re-access control ("re-acc. ctrl."), and access blocking ("acc. block") users changes with the time in the 200–300 s time period. The designed ACM module can generate

Table 2: User behavior grouping table.

	Identity authentication stage			Access control stage			Traffic detection stage		
Number of users	Normal auth. request		**100**	Normal acc. ctrl. request		**50**	Normal traffic		**25**
							Abnormal traffic	$GR_i^t < 0.4$	5
								$0.4 \leq GR_i^t < 0.7$	10
								$GR_i^t \geq 0.7$	10
				Abnormal acc. ctrl. request	$AR_i^t < 0.35$	10	—	—	—
					$0.35 \leq AR_i^t < 0.6$	15	—	—	—
					$AR_i^t \geq 0.6$	25	—	—	—
	Abnormal auth. request	$GR_i^t < 0.5$	50	—	—	—	—	—	—
		$GR_i^t \geq 0.5$	50	—	—	—	—	—	—

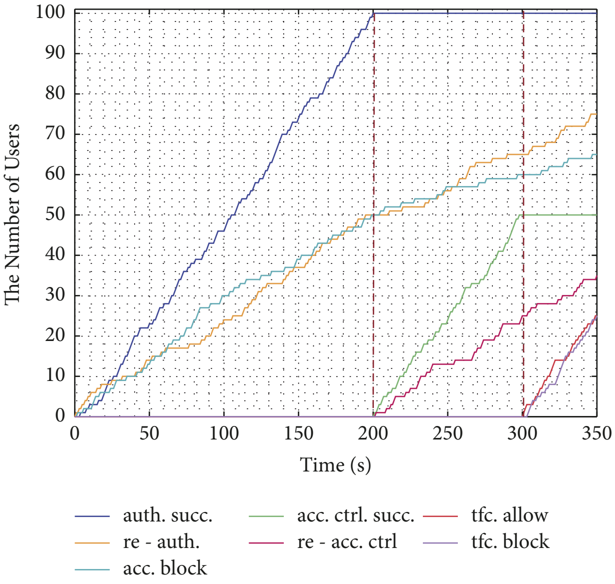


Figure 7: The evaluation of the dynamic control mechanism of the whole-process user behavior.

corresponding access control policies according to user's access action.

In the traffic detection phase, as can be seen from Figure 7, the traffic detection module can allow users who send normal traffic ("tfc. allow") to access network resources, and block the traffic sent by malicious users ("tfc. block") in time. In addition, the designed dynamic feedback mechanism can generate accurate dynamic control results ("re-auth," "re-acc. ctrl.," and "acc. block") according to the user's reputation value when the traffic detection is abnormal. When the user's reputation is lower than the threshold 0.4, the dynamic control mechanism will prevent users from accessing the network ("acc. block"). When the user reputation value is between 0.4 and 0.7, the proposed mechanism generates the dynamic control result of "re-auth." When the user's reputation is higher than 0.7, the user is asked to redo the access control process ("re-acc. ctrl.").

6. Conclusion

In this paper, we have proposed a blockchain-enabled trusted protocol based on the whole-process user behavior. The proposed WPUB-BTP constructs a trusted control chain between user identity, access action, and communication traffic, and realizes the control of user behavior in the whole process. In addition, the protocol also builds dynamic closed-loop feedback based on user reputation, which realizes dynamic control of user behavior. Eventually, we deployed the proposed protocol in the Hyperledger Fabric for evaluation. The results show that the proposed WPUB-BTP can control the whole-process user behavior and reduce the risk of network being attacked.

This paper focuses on demonstrating the dynamic trusted control mechanism based on whole-process user behavior. In future work, we will optimize the trusted subprotocol and parameter selection in each module, and conduct more in-depth research on authentication, access control, and malicious traffic detection.

Data Availability

The data that support the findings of this study can be obtained from the corresponding author upon reasonable request.

Conflicts of Interest

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

Acknowledgments

This work was supported in part by the National Key R&D Program of China, under Grant no. 2018YFA0701604, and Fundamental Research Funds for the Central Universities, under Grant nos. 2021YJS012 and 2021YJS008.

References

[1] H. Guo, J. Li, J. Liu, N. Tian, and N. Kato, "A survey on space-air-ground-sea integrated network security in 6G," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, pp. 53–87, 2022.

[2] D. Je, J. Jung, and S. Choi, "Toward 6G security: technology trends, threats, and solutions," *IEEE Communications Standards Magazine*, vol. 5, no. 3, pp. 64–71, 2021.

[3] K. Wang, W. Quan, N. Cheng, M. Liu, Y. Liu, and H. A. Chan, "Betweenness centrality based software defined routing: observation from practical internet datasets," *ACM Transactions on Internet Technology*, vol. 19, no. 4, pp. 1–19, 2019.

[4] K. David and H. Berndt, “6G vision and requirements: is there any need for beyond 5G?” *IEEE Vehicular Technology Magazine*, vol. 13, no. 3, pp. 72–80, 2018.

[5] J. Dong, K. Wang, W. Quan, and H. Yin, “InterestFence: simple but efficient way to counter interest flooding attack,” *Computers & Security*, vol. 88, Article ID 101628.

[6] X. Zhang, Y. Zhao, and G. Min, “Intelligent video ingestion for real-time traffic monitoring,” *ACM Transactions on Sensor Networks*, vol. 18, 2022.

[7] A. G Martín, A. Fernández-Isabel, I. Martín de Diego, and M. Beltran, “A survey for user behavior analysis based on machine learning techniques: current models and applications,” *Applied Intelligence*, vol. 51, no. 8, pp. 6029–6055, 2021.

[8] S. Ryu, Y. J. Kang, and H. Lee, “A study on detection of anomaly behavior in automation industry,” in *Proceedings of the 2018 20th International Conference on Advanced Communication Technology (ICACT)*, Chuncheon, South Korea, February 2018.

[9] X. Zhang, Z. Qi, G. Min, W. Miao, Q. Fan, and Z. Ma, “Cooperative edge caching based on temporal convolutional networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 9, pp. 2093–2105, 2022.

[10] S. Zhang, J. Liu, H. Guo, M. Qi, and N. Kato, “Envisioning device-to-device communications in 6G,” *IEEE Network*, vol. 34, no. 3, pp. 86–91, 2020.

[11] Y. Siriwardhana, P. Porambage, and M. Liyanage, “AI and 6G security: opportunities and challenges,” in *Proceedings of the 2021 Joint European Conference on Networks and Communications & 6G Summit*, Porto, Portugal, June 2021.

[12] L. Jin, Y. Chen, T. Wang, P. Hui, and A. V. Vasilakos, “Understanding user behavior in online social networks: a survey,” *IEEE Communications Magazine*, vol. 51, no. 9, pp. 144–150, 2013.

[13] D. Jiang, Y. Wang, Z. Lv, S. Qi, and S. Singh, “Big data analysis based network behavior insight of cellular networks for industry 4.0 applications,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1310–1320, 2020.

[14] S. Hu, Y. C. Liang, Z. Xiong, and D. Niyato, “Blockchain and artificial intelligence for dynamic resource sharing in 6G and beyond,” *IEEE Wireless Communications*, vol. 28, no. 4, pp. 145–151, 2021.

[15] A. H. Khan, N. Ul Hassan, C. Yuen et al., “Blockchain and 6G: the future of secure and ubiquitous communication,” *IEEE Wireless Communications*, vol. 29, no. 1, pp. 194–201, 2022.

[16] D. Zheng, C. Jing, R. Guo, S. Gao, and L. Wang, “A traceable blockchain-based access authentication system with privacy preservation in VANETs,” *IEEE Access*, vol. 7, Article ID 117716, 2019.

[17] Q. Feng, D. He, S. Zeadally, and K. Liang, “BPAS: blockchain-assisted privacy-preserving authentication system for vehicular Ad hoc networks,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4146–4155, 2020.

[18] C. Feng, B. Liu, Z. Guo, K. Yu, Z. Qin, and K. K. R. Choo, “Blockchain-based cross-domain authentication for intelligent 5G-enabled internet of drones,” *IEEE Internet of Things Journal*, vol. 9, no. 8, pp. 6224–6238, 2022.

[19] K. N. Qureshi, G. Jeon, M. M. Hassan, M. R. Hassan, and K. Kaur, “Blockchain-based privacy-preserving authentication model intelligent transportation systems,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–9, 2022.

[20] L. Tan, N. Shi, K. Yu, M. Aloqaily, and Y. Jararweh, “A blockchain-empowered access control framework for smart devices in green internet of Things,” *ACM Transactions on Internet Technology*, vol. 21, no. 3, pp. 1–20, Article ID 80, 2021.

[21] S. Sun, R. Du, S. Chen, and W. Li, “Blockchain-based IoT access control system: towards security, lightweight, and cross-domain,” *IEEE Access*, vol. 9, Article ID 36868, 2021.

[22] B. S. Egala, A. K. Pradhan, V. Badarla, and S. P. Mohanty, “Fortified-chain: a blockchain-based framework for security and privacy-assured internet of medical Things with effective access control,” *IEEE Internet of Things Journal*, vol. 8, no. 14, Article ID 11717, 2021.

[23] Y. Feng, W. Zhang, X. Luo, and B. Zhang, “A consortium blockchain-based access control framework with dynamic orderer node selection for 5G-enabled industrial IoT,” *IEEE Transactions on Industrial Informatics*, vol. 18, no. 4, pp. 2840–2848, 2022.

[24] M. Li, H. Zhou, and Y. Qin, “Two-stage intelligent model for detecting malicious DDoS behavior,” *Sensors*, vol. 22, no. 7, Article ID 2532, 2022.

[25] S. Cao, S. Dang, Y. Zhang, W. Wang, and N. Cheng, “A blockchain-based access control and intrusion detection framework for satellite communication systems,” *Computer Communications*, vol. 172, pp. 216–225, 2021.

[26] W. Guo, J. Xu, and Y. Pei, “A distributed collaborative entrance Defense framework against DDoS attacks on satellite internet,” *IEEE Internet of Things Journal*, vol. 9, no. 17, pp. 15497–15510, 2022.

[27] P. Ramanan, D. Li, and N. Gebraeel, “Blockchain-based decentralized replay attack detection for large-scale power systems,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 8, pp. 4727–4739, 2022.

[28] R. F. Hayat, S. Aurangzeb, M. Aleem, G. Srivastava, and J. C. W. Lin, “ML-DDoS: a blockchain-based Multilevel DDoS mitigation mechanism for IoT environments,” *IEEE Transactions on Engineering Management*, pp. 1–14, 2022.

[29] F. Guthrie, S. Lowe, and M. Saidel-Keesing, *VMware vSphere Design*, John Wiley & Sons, New York, NY, USA, 2011.

[30] Z. Tu, H. Zhou, and K. Li, “A blockchain-based user identity authentication method for 5G,” in *Proceedings of the 2021 5th International Symposium on Mobile Internet Security (MobiSec 2021)*, Jeju Island, Republic of Korea, October 2021.

[31] V. C. Hu, D. R. Kuhn, and D. F. Ferraiolo, “Attribute-based access control,” *Computer*, vol. 48, no. 2, pp. 85–88, 2015.

[32] Z. Tu, H. Zhou, K. Li, M. Li, and A. Tian, “An energy-efficient topology design and DDoS attacks mitigation for green software-defined satellite network,” *IEEE Access*, vol. 8, Article ID 211434, 2020.

[33] A. Josang and R. Ismail, “The beta reputation system,” in *Proceedings of the 15th bled electronic commerce conference*, Bled, Slovenia, June 2002.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 1761655, 12 pages
https://doi.org/10.1155/2022/1761655

Research Article

LGBM: An Intrusion Detection Scheme for Resource-Constrained End Devices in Internet of Things

Yong-Quan Cong,[1] **Ting Guan**,[2] **Ju-Fu Cui**,[3] **and Xiang-Guo Cheng**[4]

[1]*Qingdao Academy of Recruitment and Examination, Qingdao, China*
[2]*Educational Equipment & ICT Center, Qingdao, China*
[3]*Bank of Qingdao, Qingdao, China*
[4]*College of Computer Science and Technology, Qingdao University, Qingdao, China*

Correspondence should be addressed to Xiang-Guo Cheng; chengxg@qdu.edu.cn

Received 17 August 2022; Revised 19 September 2022; Accepted 28 September 2022; Published 11 October 2022

Academic Editor: Kai Wang

The intrusion detection schemes (IDSs) based on the Gradient Boosting Decision Tree (GBDT) face three problems: unbalanced training data distribution, large dimensionality of data features, and difficulty in model parameter optimization, which lead to weak monitoring capability and high false positive rate. For the problem of unbalanced training data distribution, we make the one-sided gradient oversampling algorithm to ensure the balance between the data of each category. To tackle the problem of the large dimensionality of data features, we develop a hierarchical cross-validation algorithm for binding mutually exclusive features. To address the problem of difficulty in model parameter optimization, we design a Bayesian optimization algorithm to make the model parameter search process more targeted and reduce the model training cost by establishing functional relationships between hyperparameters and target functions. The detailed experimental results show that the scheme can effectively solve the problems of data imbalance, high-dimensional data features, and low parameter finding efficiency, and improve the model's ability to monitor the attack behavior.

1. Introduction

Internet of Things (IoT) [1, 2] incorporates various types of acquisition or control sensors as well as mobile communications, intelligent analytics, and other technologies into various aspects of industrial production processes, making a large number of resource-constrained end devices gradually becoming first-class network entities [3, 4]. Compared to personal computers and cloud servers with large amounts of computing resources, end devices are usually close to the user side or in the transmission path and have a higher likelihood of being compromised by attackers. For example, an attacker can perform a side-channel attack on end devices by monitoring common information such as the time consumption and power consumption of end devices. Intrusion detection schemes are one of the most well-known security protection techniques in the traditional Internet domain [5–7]. However, since emerging resource-constrained network entities usually have limited computing power or insufficient power supply, mainstream intrusion detection techniques are hardly as effective as they were in the past. Therefore, it is necessary to design lightweight intrusion detection techniques to protect the security of resource-constrained end devices in IoT.

IDSs are mainly divided into two categories: traditional detection schemes and machine learning-based detection schemes. Traditional detection schemes suffer from weak monitoring capability [8, 9], high false positive rates [10, 11], difficult feature information collection [12], etc. To cope with these problems of traditional detection schemes, various IDSs based on machine learning [13] have been proposed one after another. These detection schemes first use machine learning algorithms to learn known attack types and then use training models to identify attacks with corresponding features. It can be broadly classified into the following two categories: (1) IDSs based on a single machine learning algorithm. Lippmann et al. [14] used a neural network composed of multilayer perceptions without hidden units to construct an anomaly detection system. The number of keyword occurrences in the Telnet session is first used as input to the neural network, and then the instances that are flagged as attacks are used as training data to train the multilayer perceptual neural network. Bivens et al. [15] used the TCP/IP data from DARPA to construct an anomaly detection system based on the multilayer perceptual neural network. It uses time windows to detect multiple packets as a group. However, the applicability of this scheme is very limited and the constructed model is simple and cannot handle large data volume, which leads to degradation of the model performance and has been rarely used in recent years. (2) IDS based on the integrated learning algorithm. Mousavi et al. [16] combined the grid search algorithm to reduce the number of input data and normal data matching the number of matches. Arif et al. [17] improved the recognition rate of the model on attack data by constructing the intrusion detection model with the help of principal component analysis unsupervised dimensionality reduction algorithm and Adaboost algorithm [18]. Nabila et al. [19] constructed a classifier by Random Forest and were able to identify four types of attacks, DOS, Probe, U2R, and R2L [20]. GBDT [21] is one of the most applied models for integrated learning to solve classification problems. IDSs [22–26] based on GBDT [27] are one of the most widely used means to defend against attacker intrusions today. However, this scheme usually requires integrated learning of multiple base models, and suffers from three problems: unbalanced training data distribution, large feature dimensionality, and difficulty in finding the optimal model parameters, which reduce the recognition accuracy, learning efficiency, and generalization ability of the model.

To solve the above problems, we propose a lightweight gradient boosting method, called LGBM, to improve the recognition accuracy, training efficiency, and generalization ability of the model. The main contributions are as follows:

(1) For the problem of unbalanced training data distribution, we develop a Gradient Borderline-synthetic Minority Oversampling Technique (GSMOTE) for expanding small samples of data (data classes with small sample size). The algorithm first updates the data samples based on the gradient value of each sample in the dataset by the unilateral gradient sampling algorithm and then uses the synthetic minority oversampling algorithm to expand the updated dataset with small samples, thus ensuring the balance among the data samples.

(2) For the problem of large dimensionality of data features, we design an Exclusive Features Binding-Hierarchy Cross-Validation algorithm (EFB-HCV) to reduce the feature dimensionality of the data. The algorithm first performs feature combinations based on the graph coloring idea and binds the mutually exclusive features existing in the data set to reduce the number of features.

(3) For the problem of difficult parameter search during model training, we propose a Bayesian Optimization algorithm (BO) to improve the optimization efficiency of model parameters. The algorithm adds a step limit to the parameter search process and regulates the search range of the parameters according to the step size, which can effectively avoid the traversal operation of all parameters.

(4) To verify the effectiveness of LGBM, we compare LGBM, Random Forest, Adaboost, Decision Tree, and GBDT with the help of four metrics: precision, recall, F-measure, and Roc curve. Detailed experimental results show that the new scheme improves the recognition rate of the model for a few attack types, the efficiency of the model parameter search, the learning efficiency, and the generalization ability.

2. Basic Knowledge

This section introduces two aspects of gradient boosting Decision Tree and basic optimization solution.

2.1. Gradient Boosting Decision Tree. GBDT is an efficient regression problem-solving method based on the boosting algorithm, which uses the regression tree as the basic classifier and a gradient boosting learning algorithm to iteratively generate a Decision Tree. Boosting algorithm is a weighted linear combination of multiple weak learners, i.e., $f(\vec{x}) = f_M(\vec{x}) = \sum_{m=1}^{M} h_m(\vec{x}, \Theta_m)$, $\vec{x}$ is the input of model, $h_m(\vec{x}, \Theta_m)$ denotes the mth model, Θ_m are the parameters of the mth model, M is the number of base models. The CART tree applied by the boosting algorithm is a binary decision tree. The CART tree generates a classification decision tree, if the data to be predicted is discrete, and a regression decision tree if the data to be predicted is continuous. As an improved algorithm of GBDT, the LGBM also uses the CART regression tree as the base learner to find the best division point containing all features. The CART regression tree uses the squared error as the discriminant of the best division point, and the regression boosting tree process is described as follows:

For the training set $\mathbb{N} = \{(\vec{x}_1, \tilde{y}_1), (\vec{x}_2, \tilde{y}_2), \ldots, (\vec{x}_N, \tilde{y}_N)\}$, the final output regression boosting tree model $f_M(\vec{x})$.

(1) Initialize regression lift tree $f_0(\vec{x}) = 0$;

(2) For the number of training sessions $m = 1, 2, \ldots, M$ there are:

   (a) calculation of residuals $r_{mi} = -[\partial L(\tilde{y}_i, f(\vec{x}_i))/\partial f(\vec{x}_i)]_{f(\vec{x})=f_{m-1}(\vec{x})}$;
   (b) construction of training residual sets $R_m = \{(\vec{x}_1, r_{m,1}), (\vec{x}_2, r_{m,2}), \ldots, (\vec{x}_N, r_{m,N})\}$;
   (c) the residuals were fitted by learning regression boosting tree r_m to obtain $h_m(\vec{x}, \Theta_m)$;
   (d) update $f_m(\vec{x}) = f_{m-1}(\vec{x}) + h_m(\vec{x}, \Theta_m)$;

(3) Obtain the final regression lift tree model $f_M(\vec{x})$, $f_M(\vec{x}) = \sum_{m=1}^{M} h_m(\vec{x}, \Theta_m)$.

2.2. Basic Optimization Solution. GBDT needs to traverse all features of all samples when constructing a Decision Tree to obtain effective splitting nodes to obtain the maximum information gain points. However, when the number of samples is large and the dimensionality of sample features is too high, its training efficiency will be significantly reduced. The new algorithm LGBM uses one-sided gradient sampling algorithm and mutually exclusive feature binding algorithm to reduce the number of training samples and the number of sample features in the training process to improve the training speed of the model.

2.2.1. One-Sided Gradient Sampling. One-sided gradient sampling is a common processing algorithm when the dataset contains a large amount of sample data. Instead of using weight values to measure the importance of the samples in GBDT, the negative gradient of the loss function is fitted. The larger the sample prediction error, the larger the absolute value of the gradient, and the worse the learning of the sample. And, the smaller the sample prediction error, the smaller the absolute value of the gradient, and the better the learning of the sample.

The one-sided gradient sampling algorithm measures the importance of the sample by the gradient of the sample, i.e., the higher the absolute value of the gradient of the sample, the higher the importance of the sample. One-sided gradient sampling keeps all samples with larger gradient values, while random sampling is performed among samples with smaller gradient values. The specific process is that firstly, the samples are arranged in descending order according to their absolute values of gradient, and then the top $a\%$ of them are selected as the large gradient sample point set A, and the remaining sample set is randomly selected $b\%$ as the small gradient sample set B. Finally, the two sets are combined and the model is trained under the updated data set.

The variance gain in the one-sided gradient sampling algorithm is defined as:

$$\widetilde{V}_{j|T}(d) = \frac{1}{n_T}\left[\frac{\left(\sum_{\vec{x}_i \in \alpha_l} g_i + 1 - a/b \sum_{\vec{x}_i \in \beta_l} g_i\right)^2}{n_{l|T}(d)} + \frac{\left(\sum_{\vec{x}_i \in \alpha_r} g_i + 1 - a/b \sum_{\vec{x}_i \in \beta_r} g_i\right)^2}{n_{l|T}(d)}\right]. \tag{1}$$

n_T is the number of all retained samples, $n_{l|T}(d)$ is the number of samples in the left subtree, and $n_{r|T}(d)$ is the number of samples in the right subtree. α_l and α_r are the set of samples with larger retention gradient values in the left subtree and right subtree. β_l and β_r are the set of samples with smaller retention gradient values in the left subtree and right subtree. The algorithm defines the approximation error as: $\varphi(d) = |\widetilde{V}_{j|T}(d) - V_{j|T}(d)|$, the gradient values are $\overline{g}_l(d) = \sum_{\vec{x}_i \in L} g_i / n_{l|T}(d)$, $\overline{g}_r(d) = \sum_{\vec{x}_i \in R} g_i / n_{r|T}(d)$. The approximation error satisfies $\varphi(d) \leq \mathbb{C}_{a,b}^2 \lambda * \max\left\{1/n_{j|T}^j(d), 1/n_{r|T}^j(d)\right\} + 2\ D * \mathbb{C}_{a,b}\sqrt{\lambda/n}$. $\mathbb{C}_{a,b} = 1 - a/\sqrt{b}\max_{x_i \in B}|g_i|$ is the maximum gradient weighted value in B. $D = \max(\overline{g}_l(d), \overline{g}_r(d))$ is selected as the maximum of the mean gradient values in the left and right subtrees. The one-sided gradient sampling algorithm increases the diversity of the base model, which helps to improve the generalization ability of the integrated model and also improves the ability of the model to monitor the attack behavior.

2.2.2. Mutually Exclusive Feature Binding. For the sample $\vec{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,n})^T$ in the dataset $\mathbb{R} = \left\{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \ldots, (\vec{x}_n, y_n)\right\}$, if for each sample $i = 1, 2, \ldots, N$, there will be no $x_{i,j} \neq 0$, $x_{i,k} \neq 0$, then the sample features j and k are mutually exclusive features. The mutually exclusive feature finding process is described as follows (Algorithm 1):

Benefiting from the histogram algorithm, LGBM first groups consecutive features into n integers (n integers represent n histograms), then iterates through the sample features, merges the features belonging to a certain integer range into the histogram represented by that integer, and finally merges each feature into each histogram, thus finding the best segmentation point based on the discrete value of the histogram.

3. Lightweight Gradient Boosting Method

This section focuses on three aspects: unbalanced data processing, data characterization, and parameter optimization.

3.1. Unbalanced Data Processing. Through the analysis of the collected intrusion detection dataset, we found that the data distribution of each category in the original dataset is extremely unbalanced, and the number of DOS attack types in the KDD dataset is about 400,000, accounting for about 80% of the total data, while the number of U2L attack types is about 60, accounting for less than 1%. This problem is likely to cause the learner to overfit large samples (data categories with large sample data) and underfit small samples (data categories with small sample data), leading to a decrease in the accuracy of model recognition. To solve this problem, we propose the Gradient Borderline-synthetic Minority Oversampling Technique (GSMOTE) to expand the small sample data.

The GSMOTE algorithm manually synthesizes new sample data for the few classes of samples that are at the boundary, which effectively solves the problem that the boundary instances are prone to misclassification in the SMOTE. Using the gradient sampling algorithm decreases the amount of sample data and reduces the amount of learning required by the model. According to the fitting principle of GBDT to the objective function, the gradient is crucial information to measure the fitting effect of samples. Therefore, we update the dataset with the gradient value of samples. If the gradient value of a sample is small and its learning error is small, it indicates that the sample has been well trained and the instances with small gradient values can be appropriately deleted in the dataset. If the gradient value of a sample is large, it indicates that the sample has not been fully learned and the instances with large gradient values in the dataset are retained.

We first update the dataset using the one-sided gradient sampling algorithm to eliminate sample points with small gradient values, and then perform data balancing on the updated dataset using the GSMOTE Algorithm 2, the pseudo-code for the GSMOTE is as follows:

Where T is the number of minority samples, k is the number of nearest neighbors, and N is the sampling rate. *MinoritySam*[][] is an array for original minority samples, *Newindex*[][] is the number of synthetic samples generated, *SyntheticSam*[][] is the array for synthetic samples, and *numattrs* is the number of attributes.

3.2. Data Feature Dimensionality Reduction.

The recursive feature elimination algorithm, which first assigns weights to each feature of the sample, is trained on the specified dataset using the base model. Then the feature weights of the trained model are extracted and the sample features with the smallest weight are removed by sorting them from largest to smallest according to their absolute values. Finally, the process is recursively repeated until the desired number of features is reached. In the sparse feature space, many features are mutually exclusive, i.e., they never obtain nonzero values at the same moment, and mutually exclusive features can be bundled into one feature. The algorithm reduces the data feature dimensionality to some extent and retains the valid feature information by cross-validation methods. However, this algorithm is costly in the process of dimensionality reduction and requires iterative training of the base model to traverse all sample features before the final sample dataset can be obtained. In addition, the sample features eliminated by recursion also contain information useful for the classification samples, and the dataset after multiple recursions will lose some effective features. Therefore, the performance of the model trained with this dataset is reduced.

The number of samples in the original dataset is large, and the feature dimension is large. According to the analysis of the original dataset, it is known that the high-dimensional data feature space has multiple features whose values will not be nonzero at the same time, i.e., the high-dimensional data feature space is sparse, which we call mutually exclusive features. Therefore, the mutually exclusive features can be used to merge multiple features in the dataset, thus reducing the number of features and the dimensionality of the features. According to the idea of the graph coloring problem, the mutually exclusive feature merging algorithm uses graph vertices to represent sample features, and there is no connection between mutually exclusive features, so when the graph is colored with K colors, there are K groups of mutually exclusive features in the graph. The pseudo-code of the Exclusive Features Binding Algorithm 3 is as follows:

The mutually exclusive feature merging algorithm performs feature combination based on the graph coloring problem idea, where features are used as vertices of a graph, edges connect two nonmutually exclusive features, and the weights of the edges indicate the total conflict values of the two features, and feature points of the same color in the graph are mutually exclusive features. For incomplete mutually exclusive features, the algorithm allows lower conflicts, so the features can be further combined to reduce the number of features and improve computational efficiency. To ensure that the original features are successfully separated after each feature combination is merged, i.e., the original feature values can be identified in the merged feature combination, the algorithm sets offsets for the corresponding feature values, appropriately changes the range of feature values, and assigns different feature values to different bins in the feature combination, thus avoiding feature value confusion after feature fusion.

To avoid the uneven distribution of data categories by the K-fold cross-validation method, we propose to use the Hierarchy Cross-Validation Algorithm (HCV), which treats the sample data of each attack category in a balanced way and uses a hierarchical data extraction method to ensure the equal proportional division of attack categories in the training and test sets. The pseudo-code of the Exclusive Features Binding-Hierarchy Cross-Validation Algorithm 4 (EFB-HCV) is as follows:

The EFB-HCV algorithm first optimizes the data feature reduction scheme of RFE-HCV to avoid recursively manipulating the dataset, and then uses the same hierarchical cross-validation method in data slicing to ensure equal proportional distribution of attack categories in the training and test sets. The specific details are described as follows:

(1) In terms of feature optimization, the EFB-HCV algorithm uses the mutually exclusive feature binding technique to feature the dataset and merges the same color feature points, i.e., mutually exclusive features, and sets an offset for incomplete mutually exclusive feature values to further reduce the number of features. The algorithm does not need to assign a weight value to each feature, which avoids the iterative training of the model and reduces the model training cost.

(2) In terms of data assignment, the data features are processed by the mutually exclusive feature binding algorithm. Then the hierarchical cross-validation algorithm divides the data proportionally. In other words, the data in each training set belong to different attack categories, and the proportion of attack categories in each training set and test set is the same as the original training set.

3.3. Parameter Optimization. The grid search algorithm first combines the values of multiple parameters and grids them, then uses each set of parameter combinations for base model training, and finally selects the best parameter combination based on the performance of the model. The updated grid search algorithm improves the efficiency of parameter search to some extent by stepping updating strategy, but both algorithms simply search for parameter combinations without making full use of the information of search points, which reduces the quality and efficiency of parameter search. Bayesian Optimization Algorithm (BO) in the case of a large number of parameter combinations can be more efficient than grid search by establishing the proxy function through finite iteration, making full use of the searched parameter information, and determining the optimal parameter combination directly based on the maximum value of the proxy function. The pseudo-code of BO is as follows (Algorithm 5):

The base model root node is initialized, the constant values are predicted, and the parameters such as model n_estimator are estimated. Where$f$is the black box function being optimized, X is the parameter search space, S is the collection function, and M is the agent model. According to the Bayesian optimization idea, firstly initialize the data set *Data* which contains n candidate solutions. Second, the n candidate solutions found by this point set are used to build a Gaussian regression model for making the posterior probabilities of other candidate points. Then, the collection function is constructed based on the posterior probabilities to find the next point that may produce the extreme value. Finally, the point that makes the function reach its maximum value is selected as the parameter of the training model.

Bayesian modeling of the function values of the black-box function using a Gaussian process gives the probability distribution of each function value, lets the function value at each point be a random variable, and multiple random variables form a random vector obeying a normal distribution. For the function $f(x)$, there are n sampling points $(x_1, x_2, \cdots, x_n)$, the corresponding function values $f(x) = [f(x_1), f(x_2), \ldots f(x_n)]$ of which form a vector, which obey a normal distribution in the Gaussian regression process:

$$f(x) \sim N\left(\mu(x_{1:n}), \sum(x_{1:n}, x_{1:n})\right). \tag{2}$$

$\mu(x_{1:n})$ is the mean vector of the Gaussian distribution. $\sum(x_{1:n}, x_{1:n})$ denotes the covariance matrix. The covariance matrix is usually implemented using a kernel function, which is defined in the Gaussian regression process as:

$$k(x_1, x_2) = \partial \exp\left(-\frac{1}{2\varepsilon^2}\|x_1 - x_2\|^2\right). \tag{3}$$

∂, ε is the parameter of the kernel function, and the mean vector is calculated from the mean function $\mu(x)$. According to the multidimensional normal distribution from the covariance matrix and the mean the vector,we can predict the probability distribution of the function value of the pointx_{n+1}, after adding the point the function value vector distribution is $f(x_{1:n+1})$ and obeys the $n+1$ dimensional normal distribution.

$$\begin{bmatrix} f(x_{1:n}) \\ f(x_{n+1}) \end{bmatrix} \sim N\left(\begin{bmatrix} \mu(x_{1:n}) \\ \mu(x_{n+1}) \end{bmatrix}, \begin{bmatrix} K & k \\ k^T & k(x_{n+1}, x_{n+1}) \end{bmatrix}\right), \tag{4}$$

where $f(x_{1:n})$ obeys the n-dimensional normal distribution, the mean vector is $\mu(x_{1:n})$, k is denoted as $k = [k(x_{n+1}, x_1),\ k(x_{n+1}, x_2), \ldots k(x_{n+1}, x_n)]$, calculated from the kernel function. The covariance matrix K can be calculated based on the mean function and the covariance function. The mean and variance expressions of the conditional distribution obeyed can be introduced according to the rules for calculating the $f(x_{n+1})$ multidimensional normal distribution as:

$$\begin{aligned} \mu &= k^T K^{-1}(f(x_{1:n}) - \mu(x_{1:n})), \\ \sigma^2 &= k(x_{n+1}, x_{n+1}) - k^T K^{-1} k. \end{aligned} \tag{5}$$

Suppose the mapping relationship between the parameters to the objective function is $f(x)$, $f(x)$ is uncertain, and the acquisition function constructed by the mathematical expectation of $f(x)$ does not satisfy the conditions of the corresponding function, i.e., the value of the acquisition function is small at the existing adopted points, and the value of the acquisition function is large at the points within the confidence interval and the mean value of the function is larger.We improve theexpectation acquisition function as follows: let n candidate solutions have been searched and the function is maximal:

$$\widehat{f}_n = \max(f(x_1), f(x_2), \ldots, f(x_n)). \tag{6}$$

Calculate the function value for the next candidate point x_{n+1} as $f(x_{n+1})$, if $f(x_{n+1}) \geq \widehat{f}_n$, then the extreme value of the function at n+1 is $f(x_{n+1})$, and vice versa $\widehat{f}_n$. After adding new candidate points, the improvement value of the function is $[f(x_{n+1}) - \widehat{f}_n]^*$, and the optimization goal is to find the candidate point x that makes the maximum improvement value.

$$EI_n(x) = E_n\left[\left[f(x_{n+1}) - \widehat{f}_n\right]^*\right]. \tag{7}$$

$E_n[*] = E[* \,|x_{1,n}, y_{1,n}]$ denotes the expected value calculated from the first n sampling points and their function values. Because the Gaussian process $f(x)$ obeys a normal distribution, located at point x the mean value is $\varphi = \varphi(x)$, and the variance is $\sigma^2 = \sigma^2(x)$, such that $\lambda = f(x)$, is introduced:

$$\begin{aligned} EI_n(x) &= \int_{-\infty}^{+\infty} \left[\lambda - \widehat{f}_n\right]^+ \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(\lambda - \varphi)^2}{2\sigma^2}\right) dz \\ &= \int_{\widehat{f}_n}^{+\infty} \left(\lambda - \widehat{f}_n\right) \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(\lambda - \varphi)^2}{2\sigma^2}\right) dz. \end{aligned} \tag{8}$$

Based on the points for dollars we get,

$$EI_n(x) = \int_{\widehat{f}_n}^{+\infty} \left(\lambda - \widehat{f}_n\right) \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(\lambda - \varphi)^2}{2\sigma^2}\right) dz$$
$$= \left(\varphi - \widehat{f}_n\right)\left(1 - \vartheta\left(\frac{\left(\widehat{f}_n - \varphi\right)}{\sigma}\right)\right) + \sigma\tau\left(\frac{\left(\widehat{f}_n - \varphi\right)}{\sigma}\right). \tag{9}$$

$\tau(x)$ is the probability density function of the standard normal distribution. $\vartheta(x)$ is the distribution function of the normal distribution. Let $\Delta(x) = \varphi(x) - \widehat{f}_n$ then,

$$EI_n(x) = [\Delta(x)]^+ + \sigma(x)\tau\left(\frac{\Delta(x)}{\sigma(x)}\right) - |\Delta(x)|\vartheta\left(\frac{\Delta(x)}{\sigma(x)}\right). \tag{10}$$

The expectation improvement function defines the expected value at each point as a function of that point, and eventually, the next candidate point is obtained based on the extreme value of the expected improvement function:

$$x_{n+1} = \arg\max EI_n(x). \tag{11}$$

4. Experimental Section

We validate the model optimization scheme for LGBM by using the Anaconda integrated development tool. First, we introduce the NSL-KDD. Secondly, we compare LGBM with GBDT, Adaboost, Decision Tree, and Random Forest under three metrics of precision, recall, F-measure, and Roc curve to elaborate advantages of LGBM. Then we analyze the effectiveness of LGBM for identifying different attack types using Roc curves. Finally, the optimization process of Bayesian and grid search for hyperparameters is compared and analyzed to verify that the Bayesian optimization algorithm has a better optimization effect on hyperparameters while ensuring optimization efficiency.

4.1. NSL-KDD Dataset. To address the problems of redundant records and unbalanced attack categories in the KDD, the NSL-KDD removes duplicate records from the training and test sets to ensure that the classifier does not bias towards a larger number of attack types, which in turn improves the detection accuracy of the classifier. Setting the number of records in the training and test sets can reduce the running cost of the experiment and eliminate the need to randomly select some data. As shown in Table 1, NSL-KDD contains four attack types (Dos, Probe, U2R, and R2L) and 21 specific attack instances, which are more abundant compared with the KDD, and the test set contains new samples of attack instances to better evaluate the classification performance of the learner. The distribution of the NSL-KDD data set is shown in Table 2.

5. Comparative Analysis of Model Recognition Performance

To test the recognition performance of LBGM to identify different classes of attacks, we compare the LGBM model with four models, GBDT, Adaboost, Decision Tree, and Random Forest under the three metrics of precision, recall, and F-measure. GBDT (Gradient Boosting Decision Tree) [27] is an iterative decision tree algorithm, which constructs a set of weak learners and accumulates the results of multiple decision trees as the final prediction output. Random forest [28] is a commonly used machine learning algorithm, which combines the output of multiple decision trees to reach a single result. The five models are trained and learned under the NSL-KDD training set, and the models are validated by the test set to derive the recognition performance of each model for the attack types under different metrics.

Figure 1(a) indicates the precision of the five models for the attack instances. From this table, it can be seen that the LGBM has a high precision for each attack type and most of the models have good precision for two common attack types, Probe and DOS. LGBM has the highest accuracy of precision compared to the other two options, which is only slightly weaker in Probe. The number of weak classifiers in the Adaboost is hard to set, resulting in a lower precision for the Probe attack type. For the two minority attack types U2R and R2L, the LGBM and the GBDT have high precision for both attacks, which is because both models use the boosting mechanism to integrate multiple weak learners to obtain strong learners, thus optimizing the overall performance of the model. Also, the LGBM uses the gradient synthesis minority class over adoption algorithm to process the data set and reasonably increases the number of samples of both U2R and R2L attacks, making the model equally good at identifying minority attack types. Decision Tree and Random Forest do not sample the dataset, resulting in a lower precision for U2R attack samples than GBDT and LGBM, but the Decision Tree is more sensitive to anomalous samples, so the Decision Tree has better precision for R2L.

Figure 1(b) presents the recall of the five models for different attack types. Compared to the precision rate, the recall of all models for U2R and R2L attack samples decrease to some extent, but the LGBM model still maintains relatively high recall for both attack samples, indicating that the model throws to maintain a good recall of positive example samples while ensuring the precision rate. For common attack types such as Probe and Dos, the number of sample features is large and the model can maintain a stable recall through training, so most of the models still present high recall for common attack types.

Figure 1(c) shows the F-measure of the five models for different attack types. A comparison of the F-measure of the five models for different attack types shows that the overall performance of the Adaboost model is poor, which is because that the model fails to effectively process the data samples and cannot effectively identify unknown or uncommon attack types. Compared with Adaboost, the LGBM has better overall performance, especially for the detection and identification of two rare attack types, U2R and R2L. That is because the sample processing of the dataset removes the sample data with small gradient values and expands the small samples (a small number of attack samples) so that the model can fully learn from them. Random Forest outperforms Adaboost in overall performance but is less effective in

```
        Input: $\mathbb{R} = \{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \ldots, (\vec{x}_n, y_n)\}$, the conflict threshold $K$;
        Output: the set **F** of feature grouping *bin*;
Step 1: Initialize *FN as* an array consisting of the number of nonzero eigenvalues;
Step 2: Iterate over all features $j$ of all samples and obtain the nonzero value $FN_j$ of $j$, sort the number of vertex features in descending
        order according to the array *FN*, and initialize the set **F**;
Step 3: Assume that the current vertex is $j$, traverse the feature grouping *bin*, calculate the conflict value *con* between $j$ and the feature
        points in the *bin*, *con* is less than $K$, then it indicates that vertex $j$ and the feature points in the *bin* do not conflict, add $j$ to the
        feature grouping *bin*;
Step 4: If vertex $j$ conflicts with the features in *bin* and is not added to the feature grouping *bin*, create a new feature grouping for that
        vertex and add it to the set **F**.
```

Algorithm 1: Mutually exclusive feature search algorithm.

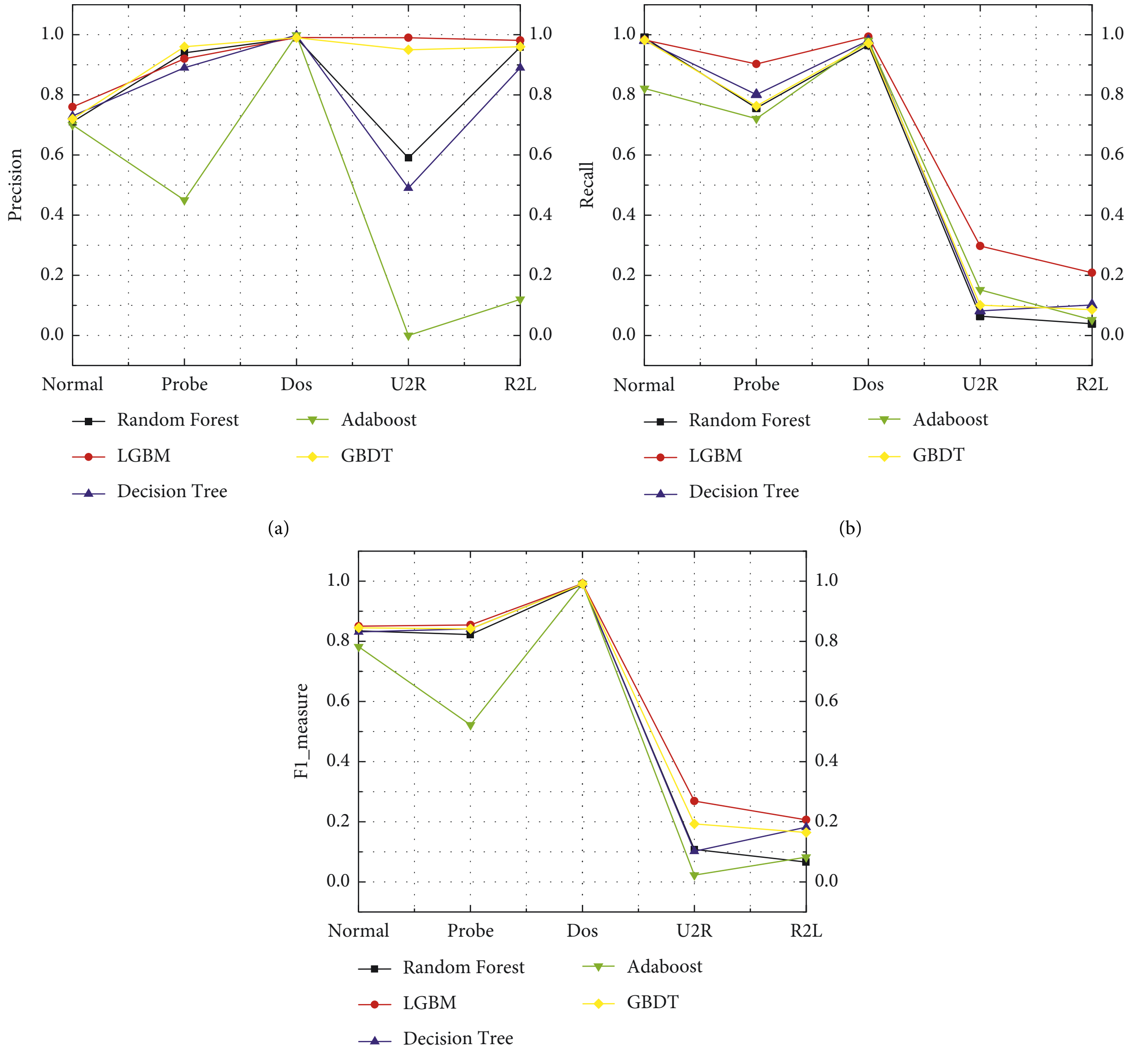
```
Initialize $T$, $k$, $N$;
**If** $N < 100$ **then**
    Randomize the $T$ minority class samples;
    $T = (N/100) * T$;
    $N = 100$;
**End if**
**For** $i = 1$ to $T$:
    Compute $k$ nearest neighbors for $i$ and save the indices in the *mArray*;
    **while** $N! = 0$ **do**
        Choose a random number $s$ between 1 and $k$;
        **for** $a = 1$ to *numattrs*:
            $dif = MinoritySam[mArray[s]][a] - MinoritySam[i][a]$;
            $gap$ = random a number between 0 and 1;
        **end for**
        *Newindex++*;
        $N = N - 1$;
    **End while**
**End for**
```

Algorithm 2: GSMOTE algorithm.

identifying and detecting uncommon attack types than LGBM models. That is because the trained models are slightly less targeted due to the lack of expansion of the small sample data. To sum up, the overall performance of the LGBM model is better than the other models.

5.1. Comparative Analysis of Model Roc Curves. Figure 2 depicts the Roc curves of the five models for different attack types. According to the definition of Roc curves, the special points in the figure are first analyzed. The point (0, 1), i.e., TPR = 1 and FPR = 0, indicates that the classifier classifies all samples correctly. The point (1, 0), i.e., TPR = 0 and FPR = 1, indicates that the classifier misclassifies all samples and has the worst performance. The two points (0, 0) (1, 1) indicate that the classifier predicts negative samples and positive samples. The ability of the model to detect and identify each attack type is known from the Roc curve of each model.

Figures 2(a) and 2(c) show that the Roc curves of Decision Tree and Random Forest have similar recognition effects on attack samples such as U2R and R2L. Also, the curves are close to $y = x$ indicating that the model classifies the samples randomly and does not effectively detect the sample data. That is because the Decision Tree and Random Forest models tend to select different attributes when classifying the category data with a large difference in the number of samples, resulting in a poor recognition rate due to insufficient training for sample data with few attributes. It can conclude that the two models have a higher recognition accuracy for a larger number of Probe, DOS, and normal data samples. Figure 2(d) shows the Roc curves of the Adaboost model for different attack types. The model appears to misclassify R2L samples and has lower recognition accuracy for other types of sample data, as the model is sensitive to the distribution of sample data and does not balance the dataset leading to a decrease in the classification accuracy of the model. The Roc curves of the GBDT and LGBM models are shown in Figures 2(b) and 2(e), and it can be seen that both of them have better recognition ability for each type of data sample. Since the GBDT model can effectively deal with anomalous data and can handle both continuous and discrete values, the model has better recognition accuracy for R2L attack types. In summary, the overall recognition accuracy of the LGBM model for different attack types is better than other models.



(a)

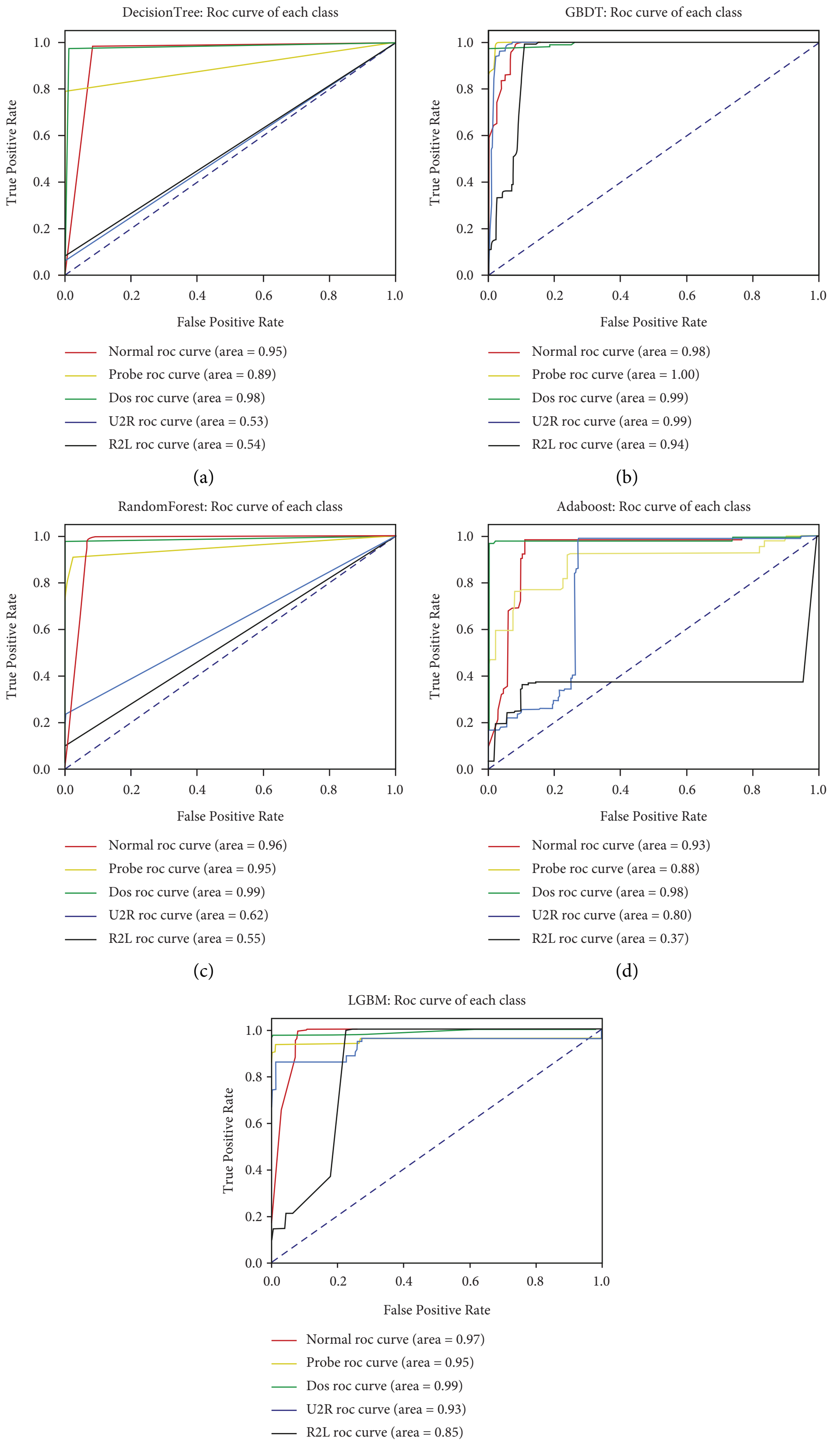
(b)

(c)

Figure 1: Accuracy of the analysis of model recognition performance. (a) Precision. (b) Recall. (c) F-measure.

5.2. Comparative Analysis of Model Parameter Optimization. Hyperparameters are set before the training of a machine learning model and directly affect the learning effect of the model. A set of optimal hyperparameters can improve the learning ability and effectiveness of the model. The grid search algorithm can iterate through all parameter combinations to find the optimal combination of parameters, but this method is less efficient. If the number of model parameters is too large, the grid search algorithm will increase the training cost of the model and reduce the efficiency of parameter optimization. Moreover, the algorithm performs an iterative search so that the model training is not targeted. Therefore, we use the Bayesian optimization algorithm to optimize the parameters, and the experimental results are shown in Tables 3 and 4. We select three important parameters, max_depth, n_estimator, and num_leaves, to measure the amount of data contained in the NSL-KDD and train them under the LGBM model with tuning parameters.

Tables 3 and 4 show the overall performance of optimization methods with different combinations of parameters. The model achieves the highest performance value of 0.9528 in the test set, the grid search algorithm, while the performance value of the model optimized with Bayesian parameters is 0.9906, which indicates that the Bayesian-optimized parameters can enhance the training learning ability of the model and obtain a stronger classifier. This is because the random combination of different parameter values by the grid search algorithm tends to lead to excessive differences between different parameter values, making it difficult for the model to reach the optimal value. The Bayesian optimization algorithm establishes a functional relationship between the hyperparameters and the model



(a)

(b)

(c)

(d)

(e)

Figure 2: Roc curve of model. (a) Decision tree. (b) GBDT. (c) Random forest. (d) Adaboost. (e) LGBM.

```
Input: F: features, MC: max conflict count, G: construct graph, NumData: number of data, B: One bundle of exclusive features;
searchOrder = G.sortbyDegree();
for i in searchOrder do:
    needNew = True;
    for j = 1 to len (bundles) do:
        cnt = ConflictCnt (bundles[j], F[i]);
        if cnt + bundlesConflict [i] ≤ MC then
            bundles [j].add (F[i]), needNew = False;
            break;
            binScope = 0, NumBin = 0;
        end if
    end for
end for
for i = 1 to NumData do
            newBin [i] = 0;
            for j = 1 to len (B) do
                if B[j].bin[i]! = 0 then
                    newBin [i] = B[j].bin[i] + binScope [j];
            end if
        end for
    end for
Output: newBin, binScope
```

Algorithm 3: Exclusive features binding.

```
Initialize Estimator, $\sum_k (\gamma_k)$, $M_1, \cdots, M_g$;
for $m = M_1$ to $M_g$ do:
    exclusive features binding for $M_i, i = 1, \cdots, g$;
    covariance matrix $Sc$ obtained from sample set;
    fitting $F(S_c, \sum_k (\gamma_k)) \longrightarrow \hat{\gamma}_{k,c}, \sum_k (\hat{\gamma}_{k,c})$;
    covariance matrix $S_V$ obtained from Validation set;
    fitting $F(S_v, \sum_k (\hat{\gamma}_{k,c})) \longrightarrow \Theta$;
    comparison of the cross-validation indices $\Theta$;
end for
Output the most stable $\Theta$.
```

Algorithm 4: EFB-HCV algorithm.

```
Initialize $f_0(x) = \text{argmin}_\gamma \sum_{i=1}^{N} L(y_i, \gamma), f, X, S, M$;
Initialize EFB-HierarchyCV;
n_estimator: Data = Samples$(f, X)$;
for $i = av$(Data) to $T$ do:
    $P(y|x, \text{Data}) = \text{Fit Model}(M, \text{Data})$;
    $X_i = \arg\max S(x, P(y|x, \text{Data}))$;
    $Y_i = f(X_i)$;
    Data = Data + $(X_i, Y_i)$;
    for $m = 1$ to $M$ do:
        for $i = 1, 2, \cdots, N$ do:
            compute $r_{im} = -[\partial L(y_i, f(x_i))/\partial f(x_i)]_{f=f_{m-1}}$;
        end for
    end for
    fit a regression tree to the targets $r_{im}$ giving terminal regions $R_{jm}, j = 1, 2, \cdots, J$;
    for $j = 1, 2, \cdots, J_m$ do:
        compute $r_{jm} = \arg\min \sum_{x_i \in R_{jm}} L(y_i, f_{m-1}(x_i) + \gamma)$;
    end for
    Update $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$;
end for
Output: $\widehat{f}(x) = f_M(x)$.
```

Algorithm 5: BO algorithm.

Table 1: Types of attacks on the NSL-KDD dataset.

Type of attack	Specific attack examples
DOS	apache2; back; mailbomb; neptune; pod; land; processtable; smurf; teardrop; udpstorm
Probe	ipsweep; mscan; portsweep; saint; satan
U2R	buffer_overflow; loadmodule; perl; ps rootkit; sqlattack; xterm; httptunnel
R2L	imap; multihop; named; phf; sendmail snmpgetattack; snmpguess; worm; xlock xsnoop; spy; warezclient; warezmaster

Table 2: The distribution of the NSL-KDD dataset.

Type	Number of records
Normal	67343 (53%)
DOS	45927 (37%)
Probe	11656 (9.16%)
U2R	52 (0.04%)
R2L	995 (0.8%)

Table 3: Parameter optimization analysis: grid search optimization.

Grid search optimization			
Target	max_depth	n_estimator	num_leaves
0.9139	8.557	188.0	59.78
0.9427	6.157	140.6	59.51
0.8709	7.764	225.1	56.0
0.9188	9.0	68.68	40.0
0.9528	6.0	10.0	40.0

Table 4: Parameter optimization analysis: Bayesian optimization.

Bayesian optimization			
Target	max_depth	n_estimator	num_leaves
0.9218	8.705	114.8	40.3
0.9432	6.084	111.0	59.57
0.9906	8.705	22.86	40.03
0.9831	8.963	12.35	40.39
0.9658	6.0	250.0	40.0

objective function, and the corresponding parameter values are obtained through the optimal value of the functional relationship.

6. Conclusion

Intrusion detection technology is one of the most well-known security protection technologies in the traditional Internet domain. However, due to the emerging resource-constrained network entities, with limited computing power or insufficient power supply, it is difficult for mainstream intrusion detection technologies to perform as effectively as before. IDSs based on GBDT face three major challenges: unbalanced training data distribution, excessive feature dimensionality, and difficulty in finding the best model parameters that cannot be effectively applied to the security protection of end devices in IoT. To solve these problems, we propose an optimization model LGBM for GBDT. Detailed experimental results verify the effectiveness of the proposed scheme.

Data Availability

All data generated or analyzed during this study are included in this article.

Conflicts of Interest

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

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of this paper. This research is supported by the National Natural Science Foundation of China (NSFC) [grant nos. 62172377, 61872205], and the Shandong Provincial Natural Science Foundation [grant no. ZR2019MF018].

References

[1] S. Wijethilaka and M. Liyanage, "Survey on network slicing for Internet pf Things Realization in 5G networks," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 957–994, 2021.

[2] Z. Cai and X. Zheng, "A Private and efficient mechanism for data Uploading in Smart Cyber-Physical systems," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 766–775, 2020.

[3] Z. Cai, X. Zheng, J. Wang, and Z. He, "Private data trading towards range counting Queries in Internet of Things," *IEEE Transactions on Mobile Computing*, p. 1, 2022.

[4] P. K. Malik, R. Sharma, R. Singh et al., "Industrial Internet of Things and its Applications in Industry 4.0: State of the Art," *Computer Communications*, vol. 166, pp. 125–139, 2021.

[5] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: a Systematic study of machine learning and Deep learning Approaches," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 1, pp. 1–29, 2021.

[6] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-Sanitization for Preventing sensitive information Inference attacks in Social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 1–590, 2016.

[7] Z. Yang, X. Liu, T. Li et al., "A Systematic Literature Review of methods and datasets for anomaly-based network intrusion detection," *Computers & Security*, vol. 116, pp. 102675–102720, 2022.

[8] T. M. Koo, H. C. Chang, Y. T. Hsu, and H. Y. Lin, "Malicious Website detection based on Honeypot systems," in *Proceedings of the 2nd International Conference on Advances in Computer Science and Engineering*, pp. 76–82, Atlantis Press, July 2013.

[9] J. Gu and S. Lu, "An effective intrusion detection Approach using SVM with Naïve Bayes feature Embedding," *Computers & Security*, vol. 103, no. 3, pp. 102158–102219, 2021.

[10] I. F. Kilincer, F. Ertam, and A. Sengur, "Machine learning methods for Cyber security intrusion detection: Datasets and Comparative study," *Computer Networks*, vol. 188, pp. 107840–107916, 2021.

[11] G. Andresini, A. Appice, and D. Malerba, "Autoencoder-based Deep metric learning for network intrusion detection," *Information Sciences*, vol. 569, pp. 706–727, 2021.

[12] J. Liu, Y. Gao, and F. Hu, "A Fast network intrusion detection system using Adaptive synthetic oversampling and Lightgbm," *Computers & Security*, vol. 106, pp. 102289–102316, 2021.

[13] O. Alkadi, N. Moustafa, B. Turnbull, and K. K. R. Choo, "A Deep Blockchain Framework-Enabled Collaborative intrusion detection for protecting IoT and cloud networks," *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 9463–9472, 2021.

[14] M. Sabokrou, M. Fayyaz, M. Fathy, Z. Moayed, and R. Klette, "Deep-anomaly: fully Convolutional neural network for Fast anomaly detection in Crowded Scenes," *Computer Vision and Image Understanding*, vol. 172, pp. 88–97, 2018.

[15] N. Moustaf and J. Slay, "Creating Novel features to anomaly network detection using DARPA-2009 data set," in *Proceedings of the 14th European Conference on Cyber Warfare and Security*, pp. 204–212, Academic Conferences Limited, July 2015.

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

[17] N. Moustafa, B. Turnbull, and K. K. R. Choo, "An Ensemble intrusion detection technique based on proposed Statistical Flow features for protecting network Traffic of Internet of Things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4815–4830, 2019.

[18] Y. Yuan, G. Kaklamanos, and D. Hogrefe, "A Novel Semi-Supervised Adaboost technique for network anomaly detection," in *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pp. 111–114, ACM, November 2016.

[19] C. A. Ronao and S. B. Cho, "Anomalous Query access detection in RBAC-Administered Databases with random forest and PCA," *Information Sciences*, vol. 369, pp. 238–250, 2016.

[20] K. Liu, X. Hu, H. Zhou, L. Tong, W. D. Widanage, and J. Marco, "Feature Analyses and modeling of Lithium-Ion Battery Manufacturing based on random forest classification," *IEEE-ASME Transactions on Mechatronics*, vol. 26, no. 6, pp. 2944–2955, 2021.

[21] V. Sharma and R. N. Mir, "An enhanced time efficient technique for image Watermarking using Ant Colony optimization and Light gradient boosting algorithm," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 3, pp. 615–626, 2022.

[22] J. Chen, X. Zhang, T. Wang et al., "Fidas: Fortifying the cloud via Comprehensive FPGA-based Offloading for intrusion detection: industrial Product," in *Proceedings of the 49th Annual International Symposium on Computer Architecture*, ACM, June 2022.

[23] Q. M. Alzubi, M. Anbar, Y. Sanjalawe, M. A. Al-Betar, and R. Abdullah, "Intrusion detection system based on Hybridizing A Modified binary Grey Wolf optimization and Particle Swarm optimization," *Expert Systems with Applications*, vol. 204, Article ID 117597, 2022.

[24] A. Telikani, J. Yang, and P. Wang, "Industrial IoT intrusion detection via Evolutionary cost-sensitive learning and Fog computing," *IEEE Internet of Things Journal*, p. 1, 2022.

[25] D. Chou and M. Jiang, "A Survey on data-Driven network intrusion detection," *ACM Computing Surveys*, vol. 54, no. 9, pp. 1–36, 2022.

[26] A. Telikani, J. Shen, J. Yang, and P. Wang, "Deep learning for intrusion detection and security of Internet of Things (IoT): current analysis, challenges, and Possible solutions," *Security and Communication Networks*, vol. 2022, pp. 1–13, 2022.

[27] G. Ke, Z. Xu, J. Zhang, J. Bian, and T.-Y. Liu, "DeepGBM: a Deep learning Framework Distilled by GBDT for online prediction Tasks," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 384–394, ACM, July 2019.

[28] A. Paul, D. P. Mukherjee, P. Das, A. Gangopadhyay, A. R. Chintha, and S. Kundu, "Improved random forest for classification," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 4012–4024, 2018.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 4996427, 9 pages
https://doi.org/10.1155/2022/4996427

Research Article

Privacy-Enhanced Intrusion Detection and Defense for Cyber-Physical Systems: A Deep Reinforcement Learning Approach

Qingyuan Lin,[1] **Rui Ming**,[2] **Kailing Zhang**,[1] **and Haibo Luo**[2]

[1] *Guangxi University of Science and Technology, Liuzhou, China*
[2] *Minjiang University, Fuzhou, China*

Correspondence should be addressed to Haibo Luo; robhappy@qq.com

Received 31 August 2022; Revised 21 September 2022; Accepted 30 September 2022; Published 10 October 2022

Academic Editor: Yulei Wu

Cyber-physical systems (CPSs) will play an important role in future real-world applications through the deep integration of computing, communication, and control technologies. CPSs are increasingly deployed in critical infrastructure, industry, and homes to achieve a smart grid, smart transportation, and smart healthcare and to bring many benefits to citizens, businesses, and governments. However, the openness and complexity brought by network and wireless communication technology, as well as the intelligence and dynamic of network intrusions make CPS more vulnerable to network intrusions and bring more serious threats to human life, enterprise productivity, and national security. Therefore, intrusion detection and defense in CPS have attracted considerable attention and have become a fundamental aspect of CPS security. However, a new challenging problem arises: how to improve the efficiency and accuracy of intrusion detection while protecting user privacy during the intrusion detection process. To address this challenge, we propose a deep reinforcement learning-based privacy-enhanced intrusion detection and defense mechanism (PIDD) for CPS. The PIDD is composed of three modules: privacy-enhanced topology graphs generation module, graph convolutional networks-based user evaluation module, and the deep reinforcement learning-based intruder identification and handling module. The experimental results show that the proposed PIDD achieves excellent performance in intrusion detection accuracy, intrusion defense percentage, and privacy protection.

1. Introduction

Cyber-physical systems (CPSs) are integral and complex systems that deeply integrate computing, communication, and physical systems. They bring a number of benefits to citizens, businesses, and governments and have attracted more attention in recent years. CPS plays an important role in wide real-world applications and has been making great business impacts in various industrial sectors, such as energy, transportation, healthcare, and manufacturing. With the rapid evolution of wireless communication networks, more and more CPS subsystems are built and connected through the communication networks, which enables more and more devices to link to CPS. However, the extensive utilization of devices with security vulnerabilities and unprotected communication networks makes CPS more prone to malicious cyber attacks and intrusions [1] (see Figure 1). These cyber threats, if they cannot be detected quickly and adjust the proper response strategy, will lead to grave consequences such as equipment damage, financial losses, and public safety. Traditional intrusion detection systems, primarily designed for conventional information technology systems, are not enough for CPS since they do not take into account the physical side of CPS.

In order to overcome these security threats, a deep reinforcement learning-based privacy-enhanced intrusion detection and defense mechanism (PIDD) is proposed for CPS. Intrusion detection and defense (IDD) is one of the most important strategy for securing CPS from malicious intrusions [2–4]; it can effectively minimize or prevent the

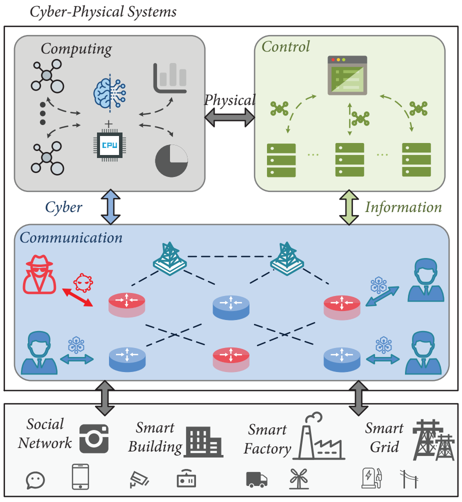


Figure 1: The architecture of cyber-physical systems and the potential security threat.

damage caused by the intrusions through performing IDD to model and monitor the malicious behaviors and intrusions early, and taking proper counter-intrusion measures and mitigation actions. With the characteristics of predicting future intrusions or security threats by building detection-based models and predictions based on empirical data, machine learning has been introduced into IDD to enhance CPS's security.

Although there are emerging machine learning-based IDD mechanisms [5–9], they do not take into account the users' privacy preservation while realizing intrusion detection and defense. Moreover, they do not combine the potential relationship between a user being an intruder and the user's communication topology graphs and features into IDD design in CPS, which helps to make the countermeasures against intrusions more efficient and reliable. The formal description of the intrusion detection problem addressed in this paper is as follows. Under the given communication conditions, it can efficiently discover intrusion behaviors and realize privacy protection at the same time.

Inspired by the previous work [10] on anomaly detection, we utilize the deep neural network with DRL training to solve the challenging problem of intrusion detection and defense in CPS. The main contributions of this paper are listed as follows.

(i) To achieve efficient intrusion detection while considering user privacy protection, we apply a variational graph autoencoder to construct a privacy-enhanced intrasystem communication topology graph and an intersystem communication topology graph with normal node characteristics. Based on these privacy-enhancing graphs and noisy node features, we employ graph convolutional networks to evaluate users' communications as regular users, intrasystem intruders, or intersystem intruders.

(ii) In order to improve the accuracy of intrusion detection, the deep reinforcement learning method twin delayed deep deterministic policy gradient algorithm (TD3) is used, which integrates the decisions made by each variational graph autoencoder during intrasystem communication and intersystem communication, respectively to determine whether the user is ultimately an intruder. Although adding noise will affect the detection accuracy, the TD3 algorithm still guarantees high-accuracy intrusion detection.

(iii) In order to effectively prevent intrusion, the corresponding countermeasures against intrusion are proposed. For intrasystem intruders, intrasystem communication is restricted, while intersystem intruders prohibit intersystem communication. In addition, both intrasystem communication and intersystem communication are prohibited for intrasystem and intersystem intruders.

(iv) Validation experiments are performed on the "CSE-CIC-IDS2018" dataset. The experimental results show that the proposed PIDD achieves excellent performance in terms of high intrusion detection accuracy, defense capability, and low privacy leakage.

The remainder of this paper is organized as follows. The proposed intrusion detection and defense framework are described in the following section. The implementation details of the DRL-based privacy-enhanced solution are then presented. Simulation results are presented and then discussed. The final section concludes this paper.

2. Overall Design of the DRL-Based Privacy-Enhanced Intrusion Detection and Defense in CPS

In this section, we first introduce the basic concept of CPS and the formulation of the intrusions and defenses problem. The proposed PIDD framework is then presented in detail.

2.1. Cyber-Physical Systems. A CPS is a controllable, reliable, and scalable multidimensional complex system that deeply integrates computing, communication, and control capabilities based on environmental perception. CPS connects physical equipment to the Internet and realizes deep integration and real-time interaction through the feedback loop of the mutual influence of computing and physical processes to add or expand new functions and detect or control physical equipment in a safe, reliable, efficient, and real-time manner. CPS enables physical devices to have five functions: computing, communication, precise control, remote coordination, and autonomy. Through the organic integration and in-depth collaboration of computation, communication, and control technologies, realtime perception, dynamic control, and information services of large-scale engineering systems are realized, which makes CPS play an important role in wide real-world applications and has been making great business impacts in various industrial sectors, such as energy, transportation, healthcare, and manufacturing.

However, the diversity of application scenarios, the openness and complexity of networking and wireless communication, and the intelligence and dynamics of intrusions bring about unpredicted security and privacy protection challenges to intrusion detection and defense mechanisms. Therefore, efficient, accurate, and privacy-enhanced intrusion detection and defense mechanisms are crucial to the success of CPS.

2.2. Intrusions and Defenses in CPS: Problem Formulation. Cyber-intrusions mainly include intrasystem intrusions and intersystem intrusions, both of which will lead to equipment damage, economic loss, public safety, and other serious consequences. Many traditional countermeasures have been proven efficient against various intrusions. For example, in [11], to authenticate user equipment, Cui et al. first developed an edge computing-enabled unified authentication framework with the consideration of privacy preservation. Then, to prevent compromised user equipments (UEs) from launching internal intrusions, they adopt reinforcement learning and design a trust evaluation-based method to detect compromised user equipment. To enhance traditional intrusion detection mechanisms, Shen et al. [12] measure the data response processing time in the interlayer, analyze network traffic to eliminate abnormal packets, and design a hybrid augmented device fingerprinting approach to eventually realize intrusion classification and detection. However, these traditional intrusion detection systems, primarily designed for conventional information technology systems, are not enough for CPS since they do not take into account the physical side of CPS.

In recent years, as one of the important strategies to protect CPS from malicious intrusions, intrusion detection and defense have been paid attention to by theoretical research and industrial applications.

In [13], a novel intrusion detection method based on network topology verification was proposed to improve the security of the controller area network with a flexible data rate network. The method reliably detected external intrusion devices through a simple random walk-based network topology construction and subsequent verification and triggered a security mode to further protect the network from attacks. To deal with intrusion detection based on dynamic data, Qi et al. [9] proposed a new anomaly detection method combining locality-sensitive hashing, isolation forest, and PCA. This method operated on multifaceted data by introducing locality-sensitive hashing and PCA, effectively captured group anomalies and could perform model updates and processe data in constant memory and time. In [14], the vulnerabilities of in-vehicle and external networks were first discussed, and a multilayer hybrid intrusion detection algorithm, including signature-based and anomaly-based intrusion detection, was proposed to detect known and unknown attacks on in-vehicle networks.

Yang et al. [15] formulated the fine-grained known/unknown intrusion detection problem as a two-stage minimization problem, where the first stage used a conditional autoencoder to seek a score metric to minimize the empirical risk of misclassifying known attacks. The second stage was to use extreme value theory to model the distribution of reconstruction errors to find another score metric to minimize the identification risk of inferring unknown attacks. To detect malicious TCP packets, Bitton and Shabtai [16] proposed a network-based intrusion detection system specifically for securing remote desktop connections. The system utilized an innovative machine learning-based anomaly detection technique for finding malicious TCP packets that carried exploits aimed at the remote desktop protocols server. High-speed networks need to process a large amount of network traffic in real time, and it is difficult to implement intrusion detection models under large amounts of big data. To process network content and build reliable machine learning-based intrusion detection models, Viegas et al. [17] proposed a new scalable and persistent intrusion detection architecture. Using deep learning and generative adversarial networks, Shu et al. [18] explored distributed SDN and designed a cooperative intrusion detection system for VANET that enabled multiple SDN controllers to jointly train a global intrusion detection model for the entire network without directly exchanging their subnetwork flows.

In fact, both communication topologies and features should be taken into account in IDD design due to the fact that the decisions made on communication features alone are not reliable. Given the difficulty of making out the specific relations between the communication topologies and the corresponding features, specific machine learning technologies, i.e., graph neural networks and deep reinforcement learning algorithms [19], should be adopted. Although machine learning technologies can efficiently detect and defend against intrusions in CPS, users might suffer from privacy leakage problems [20] due to users' data not being properly dealt with. In addition, since CPS intrusions have become more intelligent and the heterogeneity problem of CPS still exists, various domains may have specific specifications regarding the standards and objectives of security, and the IDD mechanism for one CPS domain may not match the other one.

2.3. The Proposed PIDD Framework. The framework of PIDD is shown in Figure 2, which consists of three modules: a privacy-enhanced communication topology graph generation module, a graph convolutional network-based user evaluation module, and a deep reinforcement learning-based intruder identification and processing module.

(i) *Privacy-Enhancing Communication Topology Map Generation Module.* This module first collects each user's communication topology map and features from all border routers. Then, the privacy-enhancing communication topology graph is constructed by two variational graph autoencoders (VGAE) [21] using the intrasystem and intersystem communication topology graphs, respectively. Next, appropriate noise is injected to ensure privacy protection of user communication features.

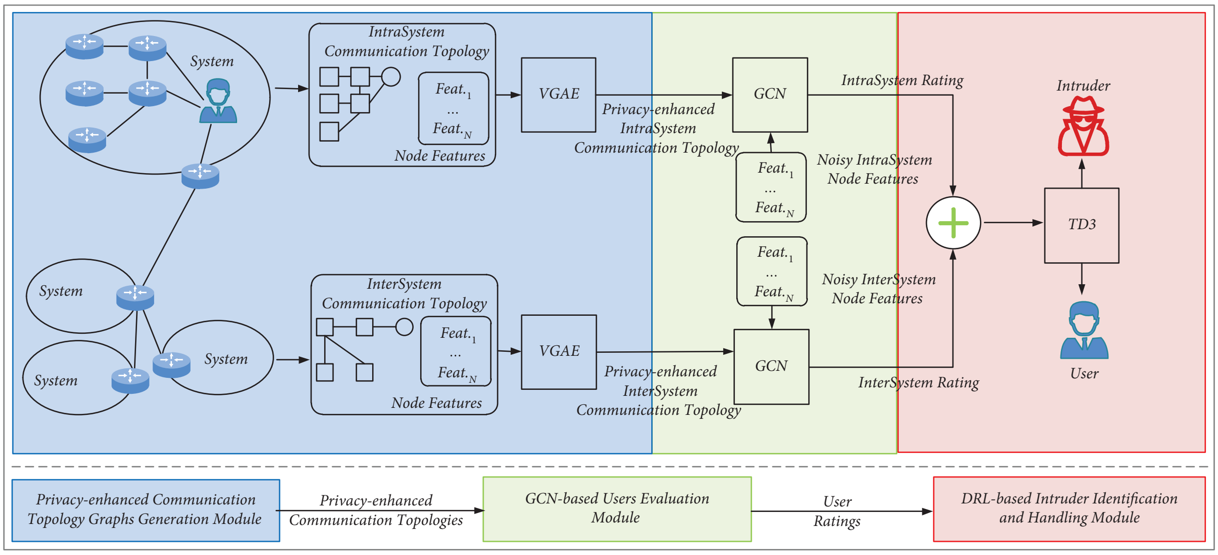


Figure 2: The framework of the proposed PIDD.

(ii) *GCN-Based User Evaluation Module*. Based on privacy-enhancing topological graphs and noisy node features, uses a graph convolutional network (GCN) [22] to evaluate users as potential regular users, intruders in the system, or intersystem intruders.

(iii) *DRL-Based Intrusion Detection and Defense Module*. This module adopts the twin delayed deep deterministic policy gradient algorithm (TD3) based on deep reinforcement learning to judge whether a user is an intruder and how to deal with different categories of users. Based on the final decision, users detected as intruders will be banned from communicating within or across systems.

In PIDD, to efficiently detect and defend intrusions, two entities of CPS, namely users and routers, are defined as follows.

(i) *Users*. There are two types of users considered in this paper. One is the normal user, while the other one is the intruder. Normal users communicate with other users within the system or cross-systems, while the intruders might launch intrusions to do different levels of damage to intrasystem routers or border routers and eventually paralyse the entire CPS.

(ii) *Routers*. For each system, there are several intrasystem routers, which coordinate the intrasystem communications, and a system border router, which is responsible for cross-system communication routing. Due to the significance of routers, to reduce the quality of service of CPS, intruders might target routers and launch the following intrusions: denial of service attacks, botnet attacks, and infiltration attacks, which are difficult to detect merely based on the features of the communication data. Within the entire CPS, the core router that coordinates all intersystems communications. We deploy the intrusion detection module on the core router to detect both intersystem intrusions and intrasystem intrusions. The intrusion detection module collects users' communication topology graphs and the corresponding features from all region-border routers for further analysis to detect intruders.

3. Models and Algorithms for the PIDD

All three modules of the proposed PIDD work collaboratively to detect and defend against intrusions in CPS, which are elaborated on in details.

3.1. Privacy-Enhanced Communication Topology Graphs Generation Module. Recall that both communication topology graphs and features of each user will be used to determine whether this user is an intruder or not. However, without proper privacy preservation, this information about users will be exposed. Injecting the proper noises can solve this problem. However, doing so will raise two other problems: (i) whether both communication topology graphs and features will be injected with noises; (ii) how much noises should be injected without causing serious detection accuracy degradation. The feasible solution to the first problem is to inject noises into communication features only. The reason for that is as follows: In order to ensure the privacy of the topology graph, the degree of each node within will be added as noise. Then, one should reconstruct the graph from the latest degree sequence. However, it is difficult due to the fact that the degree sequence might not satisfy the basic requirements of graph reconstruction [23]. Even if it is possible to reconstruct the graph, the intrusion detection accuracy will be greatly reduced, because as more communication links are added to the graph, some links will actually never exist in reality. That indicates that noise can be added to communication features only. To solve the second problem, we let each feature of the communication be normalized and added a random sampled noise from the normal distribution. Since the noise injection will result in the detection accuracy degradation, each noise is associated with a discount factor that ranges from 5% to 15%, with an increment of 5%.

We are aware that even if the communication features are protected by noise disturbance, there is always a chance that the original features will be discovered by using generative adversarial networks (GAN) [24], especially when the communication topology graphs remain the same. Thereby, we employ the VGAE to realize further privacy preservation. Basically, VGAE exploits latent variables and is able to learn interpretable latent representations for undirected graphs by using GCN as an encoder and a simple inner-product decoder. In VGAE, each communication topology graph is treated as an undirected and unweighted graph. For each graph, an adjacency matrix with diagonal elements set to 1, a degree matrix, and a random latent variable are introduced. The inference model used in VGAE is parameterized by a two-layer GCN in which both a mean vector matrix and a latent variable variance matrix are constructed. Unlike inference models, generative models are given by inner products between latent variables. VGAE takes the variational lower bound as the optimization objective of variational parameters. Note that we feed two different VGAEs with the intrasystem communication topology graphs and the intersystem communication graph of each user, respectively, to construct the individual privacy-enhanced communication topology graphs. Obviously, these two VAGEs should be trained with pairs of communication topology graphs and features of intruders or users in advance. To sum up, all VAGEs of the proposed PIDD are responsible for privacy preservation during intrusion detection.

3.2. GCN-Based Users Evaluation Module. Once privacy-enhanced intrasystem and intersystem communication topology graphs are constructed by VGAEs, we employ two GCNs to rate the user and generate the intrasystem rating and the intersystem rating, respectively.

Specifically, there are many irregular data structures. The typical ones are graph structures or topological structures, i.e., social networks, chemical molecular structures, knowledge graphs, and communication topology graphs. Similar to CNN, GCN is a feature extractor of graph data that requires both an adjacent matrix and a feature matrix so that these features can be used to classify graph data for node classification, graph classification, edge prediction, and graph embedding. In this paper, both intrasystem and intersystem intruder identification are referred by the graph classification. That suggests we can train GCNs with the labelled pairs of privacy-enhanced communication topology graphs and noisy node features about intruders constructed by using VGAEs. Once both GCNs are well trained, the GCNs' classification results about a user are considered as the intrasystem rating and intersystem rating of that user, respectively.

3.3. DRL-Based Intruders' Identification and Handling Module. It is worth mentioning that two ratings of the user, namely the intrasystem rating and the intersystem rating, cannot guarantee the user is an intruder. For example, even if both GCNs are well trained, there is always a chance that a normal user is misjudged as an intruder and vice versa due to the fact that the original communication topology graphs are altered by VGAEs, and the corresponding features are added with noises. Thereby, we introduce the overall rating of each user by calculating the weighted sum of two ratings. If the overall rating is higher than 0.5, then this user is a normal user; otherwise, the user is an intruder. To defend against intrusions, the intruder should be eliminated from the communication system of the CPS. However, some users might have overall ratings almost equal to 0.5, which might result from occasionally launching intrusions against routers in CPS. Thereby, for a user whose intrasystem rating is higher than 0.5, the intrasystem communication of this user should be forbidden; otherwise, the intersystem communication of this user should be banned.

Since the final decision is made based on the overall rating, the pair of weights should be calculated to improve intrusion detection accuracy. Note that each weight ranges from 0 to 1, with the sum of all weights equal to 1. That suggests the optimal pair of weights should be searched in a continuous space. As an off-policy method, DQN does not use the real action of the interaction each time it learns but uses the action that is currently considered to be the most valuable to update the objective value function, so there will be an overestimation of the Q value. Compared with DQN, TD3 uses two critical networks to estimate the action value function and uses soft update, policy noise, delayed learning, and gradient interception methods to solve the problem of overestimation. Thereby, we develop a twin delayed deep deterministic policy gradient (TD3) based intrusion detection mechanism. Specifically, the TD3 algorithm requires an actor-network π, a target actor network π', two critic networks Q_1 and Q_2, and their target network Q_1' and Q_2'. Basically, the network of participants chooses the action that should be taken for the state, and the network of critics evaluates this choice and prevents overestimation. We first give the definitions of state, action, and reward, respectively, as follows:

(i) *State.* Since each user might be a normal user or an intruder, let 0 represent the user being a normal one and 1 represent the user's being an intruder. Thereby, the state is constructed as a vector that consists of the binary representation of intrusion detection for all users.

(ii) *Action.* Recall that intrusion detection depends on the intrasystem rating and the intersystem rating of each user, both of which are coordinated by a pair of weights to generate the overall rating. Therefore, the pair of weights serves as the action. As the sum of two weights is equal to 1, if either weight is higher than 0.5, then the corresponding intrusion detection result is more dominant than the other. Moreover, the action should include countermeasures against intruders. If the user is an intrasystem intruder, an intersystem intruder, or both, then the user is forbidden to communicate with intrasystem users, intersystem users, or both accordingly.

(iii) *Reward*. The goal of intrusion detection is to detect and eliminate intruders to greatly reduce the number of intrusions in CPS. That suggests the intrusions prevented should be taken into account in the reward calculation. Moreover, the communication traffic should be considered as well, asimproperly chosen weights might result in a significant communication traffic drop. Thereby, we let the normal communication traffic, which equals the overall communication traffic minus the intrusion traffic, be the reward to evaluate the performance of the proposed PIDD.

In TD3 training, we randomly sample N experience to update the critic network with the loss function,

$$L\left(\vartheta^{Q_i}\right) = \frac{1}{N}\sum_{j}^{N}\left[Q_i\left(s_j, a_j|\vartheta^{Q_i}\right) - Y_j\right]^2, \tag{1}$$

where

$$Y_j = r_j + \gamma\left[Q_i'\left(s_{i+1}, \pi\left(s_{i+1}|\vartheta^{\pi'}\right)|\vartheta^{Q_i'}\right)\right]_{i=1,2}. \tag{2}$$

Thereby, we have,

$$\vartheta^{Q_i} \leftarrow \vartheta^{Q_i} - \eta\frac{\partial L\left(\vartheta^{Q_i}\right)}{\partial\vartheta^{Q_i}}. \tag{3}$$

Then, we update the actor-network π by optimizing the objective function,

$$J\left(\vartheta^{\pi}\right) = \sum_{j}^{N}\left[Q_1\left(s, a|\vartheta^{Q_1}\right)\pi\left(s_j|\vartheta^{\pi}\right)|s = s_j, a = \pi\left(s_j|\vartheta^{\pi}\right)\right]. \tag{4}$$

with

$$\vartheta^{\pi} \leftarrow \vartheta^{\pi} + \iota\frac{\partial J\left(\vartheta^{\pi}\right)}{\partial\vartheta^{\pi}}. \tag{5}$$

Next, the parameters of target networks $\vartheta^{Q'}$ and $\vartheta^{\pi'}$ are updated with a learning rate κ.

Note that the training process for all three modules of the proposed PIDD is as follows. First, VGAE and GCN are trained using all labelled communication topology maps and features of users and intruders in the privacy-enhancing communication topology map generation module and the GCN-based user evaluation module. Then, TD3 is trained using the corresponding ratings in the DRL-based intruder identification and a processing module. Once trained, the proposed PIDD is able to determine whether a user is an intruder based on the user's communication topology and characteristics.

The main symbols and their meanings for the proposed PIDD are shown in Table 1.

Table 1: Main symbols and meanings.

Symbol	Meaning
CPS	Cyber-physical systems
IDD	Intrusion detection and defense
GCN	Graph convolutional network
VGAE	Variational graph autoencoders
TD3	Twin delayed deep deterministic policy gradient
π	Actor network
π'	Target actor network
$Q_{i=1,2}$	Critic network
$Q'_{i=1,2}$	Target critic network
η, ι, κ	Learning rate

4. Numerical Results

To evaluate the performance of the proposed mechanism, we target three attacks, namely the denial of service attack (DoS), the botnet attack (Bot), and the infiltration attack (Inf), to prevent intrusion. The experiment was conducted to evaluate the performance of the proposed PIDD in Python on a computer equipped with an i7 6.4GHZ processor, 32G memory, and a win7 64-bit system. In VGAE, initialized weights are set as described in [21], and a 32-dim hidden layer and 16-dim latent variables are used in all experiments. There are up to 200 iterations of training using Adam with a learning rate of 0.01.

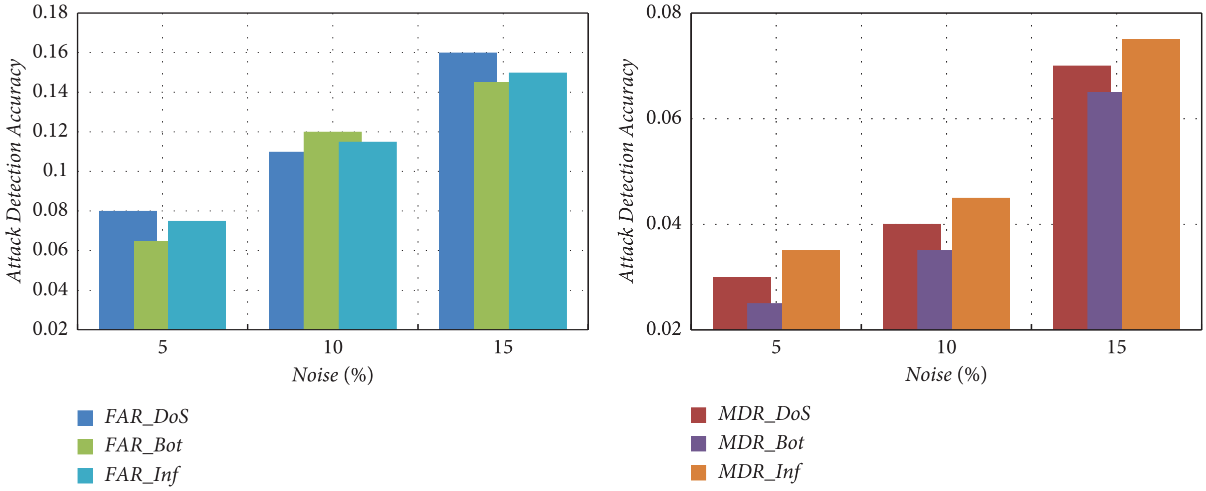
The "CSE-CIC-IDS2018" dataset, which is available at "https://www.unb.ca/cic/datasets/ids-2018.html," is used in this experiment. The dataset includes seven different attack scenarios: Brute-force, Heartbleed, Botnet, DoS, DDoS, Web attacks, and infiltration of the network from inside. The attacking infrastructure includes 50 machines, and the victim organization has 5 departments and includes 420 machines and 30 servers. The dataset includes the captured network traffic and system logs of each machine, along with 80 features extracted from the captured traffic by using CICFlowMeter-V3. To facilitate the performance evaluation, each cyber-physical system contains at most 16 routers and 1 border router. Both intrasystem communication topology graphs and intersystem communication topology graphs are extracted first. Then, all these topology graphs and communication features are used to determine whether users are intruders. The following indexes are employed to evaluate the performance of the PIDD with the consideration of different percentages of noise added.

(i) *Detection Accuracy*. Both the false alarm rate (FAR) and the miss detection rate (MDR) are applied to evaluate the detection accuracy.

(ii) *Intrusion Prevention Percentage*. The percentage of intrusions prevented in overall intrusions launched

(iii) *Privacy Preservation Percentage*. The differences between the original communication topology graphs and the privacy-enhanced ones are measured in the privacy preservation percentage.

Figure 3 shows the detection accuracy of adding different percentages of noise. As shown in Figure 3, we find that the FAR and MDR of all three types of intrusions increase as the



(a)

(b)

FIGURE 3: Detection accuracy with different percentages of noise added. (a) FAR and (b) MDR.

percentage of added noise increases. Furthermore, PIDD achieves on average 6%, 11.5%, and 14% of FAR and 3%, 4%, and 7% of MDR, and all noises are 5%, 10%, and 15% intrusion, respectively. What's more, even though the features add up to 15% noise, the highest FAR and MDR of PIDD are still lower than 16% and 8%. This is because the proposed PIDD combines the graph variational autoencoder and the graph neural network and considers intrasystem and intersystem communication at the same time, so it can effectively discover the intrusion behavior of the attacker. Experimental results show that PIDD can accurately detect routing intrusions in CPS with noisy communication data.

Table 2 gives the intrusion prevention that adds different percentages of noise in terms of intrasystem intrusion and intersystem intrusion. As observed in Table 2, it is clear that the percentage of intrusion prevention decreases with the percentage of added noise, as expected. Note that PIDD is more effective at preventing intrasystem intrusion than intersystem intrusion. This may be due to intruders launching intrasystem intrusions more frequently, making intrusion patterns harder to learn. Additionally, PIDD can detect and block at least 83% of intrasystem intrusions and 81% of intersystem intrusions, even when up to 15% of the noise is added to the communication signature. The intrusion prevention shown in Table 2 shows that PIDD can effectively defend against routing intrusion in CPS because the variational graph autoencoder and graph neural network adopted by PIDD can well capture the characteristics of intrusion behavior.

TABLE 2: Intrusion prevention with different percentages of noise added.

Intrusion	Noise		
	5 (%)	10 (%)	15 (%)
Intra_DoS	93	87	83
Inter_DoS	91	85	82
Intra_Bot	93	87	83
Inter_Bot	89	85	84
Intra_Inf	92	87	83
Inter_Inf	88	86	81

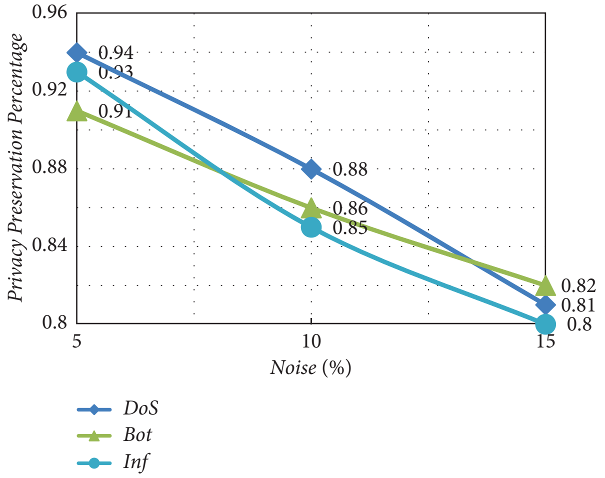


FIGURE 4: Privacy preservation with different percentages of noise added.

Figure 4 shows the privacy protection of adding different percentages of noise. It is worth mentioning that, in order to protect the privacy of users, only noise has been added to the communication function. As the percentage of added Gaussian noise increases, the user's privacy is better protected. On the other hand, VGAE modifies the user's communication topology map to a privacy-enhanced communication topology map as the input to the GCN-based classifier. Both noise injection and VGAE-based graph modification provide user privacy protection, which is verified in this figure. Furthermore, PIDD achieves on average about 91%, 86%, and 82% privacy preservation, adding 5%, 10%, and 15% of noise, respectively. This shows that PIDD can protect the privacy of users during the intrusion detection process.

5. Conclusion

In order to improve the efficiency and accuracy of intrusion detection and protect user privacy from being leaked during the CPS intrusion detection process, this paper proposes a privacy-enhanced intrusion detection and defense mechanism based on deep reinforcement learning. Specifically, first, two variational graph autoencoders are trained to generate privacy-enhanced communication topology graphs. Second, two graph convolutional networks are trained based on the privacy-enhanced communication topology map and noise features to perform user evaluation. Finally, a deep reinforcement learning algorithm TD3 is applied to identify intruders and execute appropriate countermeasures. We conducted validation experiments on the "CSE-CIC-IDS2018" dataset. Experimental results show that the proposed PIDD achieves excellent performance in terms of intrusion detection accuracy, intrusion prevention percentage, and privacy protection.

Although the proposed algorithm can perform intrusion detection under the condition of preserving privacy, the detection accuracy needs to be improved. Our future research directions include how to further combine the characteristics of intrusion behavior with the communication topology of intrusion.

Data Availability

The "CSE-CIC-IDS2018" dataset, which is available at "https://www.unb.ca/cic/datasets/ids-2018.html," is used in this experiment. We have given this site in our manuscript.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Nature Science Foundation of China (Grant numbers: 61902167, 32201679), the Science Foundation of the Fujian Province, China (Grant numbers: 2021J011015, 2022J05230), and Foundation of Fuzhou Institute of Oceanography (Grant number: 2022F07).

References

[1] S. Singh, N. Yadav, and P. K. Chuarasia, "A review on cyber physical system Attacks: issues and challenges," in *Proceedings of the 2020 International Conference on Communication and Signal Processing (ICCSP)*, pp. 1133–1138, IEEE, Chennai, India, July 2020.

[2] P. Freitas de Araujo-Filho, G. Kaddoum, D. R. Campelo, A. Gondim Santos, D. Macedo, and C. Zanchettin, "Intrusion detection for cyber–physical systems using generative adversarial networks in fog environment," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6247–6256, 2021.

[3] B. Li, Y. Wu, J. Song, R. Lu, T. Li, and L. Zhao, "DeepFed: federated deep learning for intrusion detection in industrial cyber–physical systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5615–5624, 2021.

[4] S. E. Quincozes, D. Mossé, D. Passos, C. Albuquerque, L. S. Ochi, and V. F. dos Santos, "On the performance of GRASP-based feature selection for CPS intrusion detection," *IEEE Transactions on Network and Service Management*, vol. 19, no. 1, pp. 614–626, 2022.

[5] Z. Huang, Y. Wu, N. Tempini, H. Lin, and H. Yin, "An energy-efficient and trustworthy unsupervised anomaly detection framework (EATU) for IIoT," *ACM Transactions on Sensor Networks*, 2022.

[6] O. Alkadi, N. Moustafa, B. Turnbull, and K.-K. R. Choo, "A deep blockchain framework-enabled collaborative intrusion detection for protecting IoT and cloud networks," *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 9463–9472, 2021.

[7] Y. Xie, D. Feng, Y. Hu, Y. Li, S. Sample, and D. L. Long, "Pagoda: a hybrid approach to enable efficient real-time provenance based intrusion detection in big data environments," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 6, pp. 1283–1296, 2020.

[8] H. Liu, S. Zhang, P. Zhang et al., "Blockchain and federated learning for collaborative intrusion detection in vehicular edge computing," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 6073–6084, 2021.

[9] L. Qi, Y. Yang, X. Zhou, W. Rafique, and J. Ma, "Fast anomaly identification based on multiaspect data streams for intelligent intrusion detection toward secure industry 4.0," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 9, pp. 6503–6511, 2022.

[10] X. Wang, S. Garg, H. Lin et al., "Towards accurate anomaly detection in industrial internet-of-things using hierarchical federated learning," *IEEE Internet of Things Journal*, vol. 9, no. 10, 2022.

[11] Q. Cui, Z. Zhu, W. Ni, X. Tao, and P. Zhang, "Edge-intelligence-empowered, unified authentication and trust evaluation for heterogeneous beyond 5G systems," *IEEE Wireless Communications*, vol. 28, no. 2, pp. 78–85, 2021.

[12] C. Shen, C. Liu, H. Tan, Z. Wang, D. Xu, and X. Su, "Hybrid-augmented device fingerprinting for intrusion detection in industrial control system networks," *IEEE Wireless Communications*, vol. 25, no. 6, pp. 26–31, 2018.

[13] T. Yu and X. Wang, "Topology verification enabled intrusion detection for in-vehicle CAN-FD networks," *IEEE Communications Letters*, vol. 24, no. 1, pp. 227–230, 2020.

[14] L. Yang, A. Moubayed, and A. Shami, "MTH-IDS: a multi-tiered hybrid intrusion detection system for Internet of vehicles," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 616–632, 2022.

[15] J. Yang, X. Chen, S. Chen, X. Jiang, and X. Tan, "Conditional variational auto-encoder and extreme value theory aided two-stage learning approach for intelligent fine-grained known/unknown intrusion detection," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 3538–3553, 2021.

[16] R. Bitton and A. Shabtai, "A machine learning-based intrusion detection system for securing remote desktop connections to electronic flight bag servers," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 3, pp. 1164–1181, 2021.

[17] E. Viegas, A. O. Santin, and V. Abreu Jr, "Machine learning intrusion detection in big data era: a multi-objective approach for longer model lifespans," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 1, pp. 366–376, 2021.

[18] J. Shu, L. Zhou, W. Zhang, X. Du, and M. Guizani, "Collaborative intrusion detection for VANETs: a deep learning-based distributed SDN approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4519–4530, 2021.

[19] Z. Yan, J. Ge, Y. Wu, L. Li, and T. Li, “Automatic virtual network embedding: a deep reinforcement learning approach with graph convolutional networks,” *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1040–1057, 2020.

[20] Y. Wu, H.-N. Dai, H. Wang, and K.-K. R. Choo, “Blockchain-based privacy preservation for 5G-enabled drone communications,” *IEEE Network*, vol. 35, no. 1, pp. 50–56, 2021.

[21] T. N. Kipf and M. Welling, “Variational Graph Auto-Encoders,” November 2016, https://arxiv.org/abs/1611.07308.

[22] T. N. Kipf and M. Welling, “Semi-supervised Classification with Graph Convolutional Networks,” September 2016, https://arxiv.org/abs/1609.02907.

[23] H. Huang, D. Zhang, F. Xiao, K. Wang, J. Gu, and R. Wang, “Privacy-preserving approach PBCN in social network with differential privacy,” *IEEE Transactions on Network and Service Management*, vol. 17, no. 2, pp. 931–945, 2020.

[24] C. Xu, J. Ren, D. Zhang, Y. Zhang, Z. Qin, and K. Ren, “GANobfuscator: mitigating information leakage under GAN via differential privacy,” *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 9, pp. 2358–2371, 2019.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 4748946, 12 pages
https://doi.org/10.1155/2022/4748946

Research Article

Web-Cloud Collaborative Mobile Online 3D Rendering System

Chang Liu,[1] **Huilin Song**,[2] **Ting Fang**,[1] **Qiaofeng Ou**,[1] **Geng Yu**,[1] **Tao You**,[1] **and Ming Ying**[1]

[1]*School of Information Engineering, Nanchang Hangkong University, Nanchang, China*
[2]*School of International Economics and Trade, Jiangxi University of Finance and Economics, Nanchang, China*

Correspondence should be addressed to Chang Liu; lcsszz83@gmail.com

Received 12 July 2022; Accepted 13 September 2022; Published 27 September 2022

Academic Editor: Yuanlong Cao

The collaborative online 3D rendering system proposed in this paper ensures the quality of user experience and protects online rendering resources. In this system, the conditional generative adversarial network is used to calculate complex global illumination information instead of rendering them on cloud servers. The web front-end generates high-frequency direct lighting information in real-time and displays the final result which is a blend of front-end direct lighting information and back-end indirect lighting information. Experiments show that our proposed system can improve the rendering quality of the Web3D front-end, ensure Web-Cloud load balance, and protect rendering resources online.

1. Introduction

In the era of the Internet+, with the development of Web3D technology, more and more users are accustomed to the flexibility of experiencing 3D content on various portable devices, such as mobile phones, laptops, and head-mounted devices. Web3D technology, which puts 3D content on Web browsers, is supported by most mobile devices. This technology has a revolutionary impact on the new generation of Web services and produces various critical applications in the smart city, virtual tourism, virtual museums, e-commerce, etc.

The advantages of Web3D are excellent cross-platform, but its defect is limited rendering power. Web3D system rendering capabilities are mainly determined by its core graphics application programming interface (API) “WebGL” and hardware configuration. The latest WebGL 2.0 technology version is based on OpenGL ES 3.0 designed for embedded devices. Therefore, it is difficult to achieve the same rendering performance as on the personal computer with high-level graphics API “OpenGL.” Besides, the loading latency of three-dimensional (3D) model data on the Web also poses a significant challenge, resulting in a long waiting time, which significantly reduces the user’s quality of experience (QoE). In addition, the 3D model resources in the Web3D application are directly transmitted to the Web front-end, which brings the risk of resource leakage.

For this, the collaborative rendering system has emerged to split the complex task of rendering the scene between the cloud server and the Web client. As a collaborative rendering system, our CloudBaking [1, 2] is a dedicated Web3D application-oriented dynamic scene lighting and shadow rendering system intended to compensate for the inadequate rendering capability of the Web3D application. Therefore, we design this system to perform collaborative lighting and shadow rendering at both the client and server for the Web3D scene. This system assigns the high-complexity lighting and shadow rendering to the cloud server, including soft shadow, global illumination, and so on. The web client performs the task of low-complexity renderings, such as direct lighting and screen-space ambient occlusion. Therefore, the high-precision 3D scene model is safely placed in the cloud, while the Web front-end only needs a lightweight and encrypted low-precision scene model, which reduces the risk of resource leakage.

2. Related Work

2.1. Web3D Technology. In 1997, the virtual reality markup language (VRML) was officially released as an international standard for Web3D, making it possible for 3D model files [3] to be transferred over the Internet. In August 2004, the X3D specification was released as an international standard for Web3D. X3D integrated technologies such as XML, Java, and streaming at that time in the hopes of increasing processing power, rendering quality, and speed of transmission. The Khronos Group released the WebGL 1.0 specification in March 2011. WebGL 1.0 is based on OpenGL ES 2.0 and provides APIs for 3D graphics. The WebGL 2.0 specification was released in 2013, and, based on OpenGL ES 3.0, was supported for the first time in major Web browsers such as Firefox, Chrome, and Opera [4].

In recent years, WebGL is used as a graphics engine in many Web3D applications. Furthermore, many companies have developed their own advanced rendering engines based on WebGL, such as three. js and Babylon. [4]. Among these, three. js is a 3D graphics real-time rendering library based on JavaScript and WebGL. It has become gradually favored by the majority of users because of its efficient and plug-in-free Web-side rendering capabilities. Many Web3D rendering systems use three. js, such as the Web page visualization system proposed by Marion and Jomier [5], and the real-time visualization and segmentation system of real-time visual medical images developed by Jacinto et al. [6]. Similarly, our system also uses the three. js engine.

2.2. Web3D System. Web3D systems have been widely used in people's daily life for their excellent cross-platform and easy deployment. In a Web3D system, data are transformed into visual 3D models and presented on the Web, stimulating people's interest. Virtual heritage (VH), which displays the digitization of culturally historical artifacts for display on the Web browser, is a very typical use of Web3D. Currently, website presentations cannot satisfy the extensive and intensive experience of VH users. The VH websites provide narrative knowledge, annotation experience, and mobile environment experience to adapt to the changes [7]. Building information models (BIMs) have recently become the mainstream visualizing data in the building field. To display such data with high volume, variety, velocity, and value attributes on Web browsers, researchers have created an online Web3D system based on semantic analysis and light-weighting technology [8].

Web3D technology has been widely used to build many educational virtual environments (EVEs). In the beginning, EVEs were used to create visual immersion-based virtual scene display cases, such as the Webtop system [9]. Besides, the researchers of EVEs are also concerned about user engagement, interaction, and collaboration, such as Web3D-based surgical training simulators for the treatment of trigeminal neuralgia [10], virtual space-time environments online [11], and virtual war online [12]. Users of the Web3D system can experience immersion in education, training, and tourism without leaving home at a low cost.

2.3. Web3D Rendering System. A real-time rendering system is the critical subsystem of a Web3D system, responsible for the 3D scene's loading and rendering. We classify Web3D rendering systems into three categories based on which "side" the rendering task occurs.

2.3.1. Local Rendering System. This system puts the main rendering tasks on the Web browser. The server is responsible for storing and transmitting the 3D scene's data without participating in rendering tasks. The VH system [7], virtual building system [8], and Webtop systems [9] mentioned above employ this kind of rendering system.

2.3.2. Remote Rendering System. This system puts the main rendering task on the servers and delivers the rendered results to the Web browser in the form of image steam. The web client is only used to display the rendering results without participating in rendering. This system is very suited for the Web3D application with high-quality rendering performance demand [4]. By avoiding the direct loading of 3D models on web browsers, the system can protect 3D models from illegal downloading by users [13].

2.3.3. Hybrid Rendering System. This system combines the two systems described above and allows the Web-Client side and cloud-server side to render collaboratively. Such a system avoids the waste of front-end rendering resources and guarantees the execution of expensive rendering tasks. This kind of system is widely used to study lighting rendering in dynamic scenes.

The Cloud Light system first proposed allocating lighting rendering tasks [14]. Remote Asynchronous Indirect Lighting (RADL) implements viewpoint-independent collaborative lighting rendering [15]. Shading Atlas Streaming (SAS) [16] puts all shading calculations in the cloud to complete and uses the Shading Atlas mechanism of Virtual Texture for storage. Also, the Cloud Baking system (CB system) proposed by ourselves is a typical hybrid rendering system [1, 2].

2.4. Generative Adversarial Networks. The GAN represents a deep learning model based on two sets of the neural network, generator, and discriminator, for game-based mutual learning to generate the desired output. Since the suggestion of Goodfellow et al. was proposed [17], GAN has achieved remarkable results in the areas of "image to image" translation [18, 19], style transfer [20], super-resolution [21], and 3D model generation [22, 23], etc. The research into the image-to-image translation field provides us with a direct motivation to replace the image-based pre-rendering mechanism and the real-time rendering at the cloud server with GAN. In doing so, the images generated by the cloud server will be saved in the cloud server for training the relevant GAN model, and the generative model ends up being sent to the web client to generate the global illumination map (GI map).

3. System Architecture

Based on our CloudBaking (CB) [2], we present an intelligent remote rendering system called the Intelligent CloudBaking (ICB). For completeness, we offer the whole pipeline and workflow of the ICB system and focus on the ICB system's advancement compared to the CB system. As the CB system, the ICB system also consists of two separate rendering systems: the Web-Client system and the Cloud-Server system. The protocol for connecting the two systems is also WebSocket, as shown in Figure 1.

Similar to the CB system, our system contains two modules: cloud server and client. But we newly proposed a GAN-based pre-rendering module, which transfers the pre-rendering task from the CRT-buffers manager to the GAN. This solves the problem of cloud storage space limitation, and the images stored in CRT-buffers provide a large amount of input data for GAN training. The 3D Warping technology based on the prediction mechanism is used to eliminate the hole artifacts of the generated GI map. The original 3D scene is preprocessed into LMP and then progressively streamed to the client to generate DI-map. Finally, the mixed image of DI-map and GI map is presented to the user on the client. In addition, the resources obtained by the cloud of this system, such as Light-Weight 3D Scene Steam, encoded GI-map steam, and GAN for GI maps, have been lightweight or encoded, which further improves the protection of rendering resources.

3.1. Cloud-Server Subsystem. Like the CB system, ICB lightweight the original 3D scene for reducing the initial loading time [24] and progressively streams the lightweight version to the Web-Client for rendering. To reduce the initial loading time, each 3D scene is preprocessed into LPM [24] and stored on the rendering server. The rendering server progressively streams the LPM version to the client renderer for rendering upon request from the client.

However, the cloud-sever no longer generates the GI maps only like the CB but renders a cloud rendering texture buffer (CRT-buffer) for the current scene. As shown in Table 1, the CRT-buffer contains a group of images rendered under the current viewing frustum, which can be divided into two categories, including: (1) images whose pixel values store the rendered scene with lighting and shadow information (e.g., GI map, albedo map, and direct lighting map), called L&S-images and (2) images whose pixel values record the rendered scene geometry information (e.g., the depth map and normal map), called G-images. When the CRT-buffer has been rendered in Cloud-Server, we design an octree-based CRT-buffers manager to store them rather than discard them as in the earlier system. These images stored in Cloud-Sever can be used as input data to pre-render the GI map (e.g., when the CRT-buffers store the GI maps under a similar view), or to train the GAN for GI maps offline (e.g., when the CRT-buffer manager store the input data enough). After storing many images in the CRT-buffer manager of Cloud-Server, our system pre-renders the GI map by CRT-buffers manager searching first instead of rendering it.

3.2. Web-Client Subsystem. The web client with good cross-platform and extensibility is the primary interaction medium. However, despite the continued improvement to external equipment performance that carries the Web3D applications, a compromise on the rendering capability remains necessary for the Web3D technique to embed on the Web. Therefore, to enhance the web client's interactivity while reducing the pressure from rendering, we restricted the task of direct lighting rendering to the web client and branded the results after rendering it as a direct-lighting map (DL map). Many GI maps either derived from rendering at the cloud server or generated by the neural network at the web client will be sent here at the time of scene editing by the user at the web client. This is achieved in the following steps: (1) The web client checks out whether there is a generative network locally. (2) If the generative network does not exist, then send GI map request to the cloud server and await rendered GI map transmitted from the cloud server. (3) Otherwise, GI map of the current viewpoint should be generated by the generative model at the web client. Meanwhile, the input images of the generated model need to be rendered on the Web client, and we call these maps the WRT-buffer (as shown in Table 2).

Both approaches to GI map generation will contribute to interactive latency, with the only difference being that the latency would occur at different phases. For the former one, it would occur during the stages, including the cloud server rendering stage, GI map encoding stage, and transmission stage. The latter one would happen when the GI map is generated by the generative model at the web client. Limited by the interaction delay, the rendering system with viewpoint correlation is incapable of ensuring consistency between the viewpoint in the rendered DL map and that in the generated GI map, distortion is bound to arise from blending the two maps at the web client. In the process of prediction, a camera with a larger range of field-of-view (FOV) than the web client was chosen for the cloud server, for which the GI map information rendered in the cloud server could have a wider range than the DL map information rendered at the web client, and thus the predicted probability of artificial hole generated to GI map would be reduced. As revealed in the experiment, the higher the camera's FOV value for the training set of rendering, the better the quality of a map generated by the neural network. Finally, the blending GI map and DL map at the web client (weighed averaging for pixels) could derive the rendered frame as output for ultimate display.

4. Prefetch Strategy for 3D Warping

The 3D warping method is a technique of warping a reference image to an arbitrary viewpoint by projecting pixels on the reference image plane into the 3D space and reprojecting them to the target image plane (see the Resources [2] for more details). All 3D warping methods produce hole artifacts, and our method is no exception. The hole artifacts appear in our method because 3D models' vertex cannot find the texture coordinates on a pixel-missing reference image (GI map), as shown in Figure 2. Our method

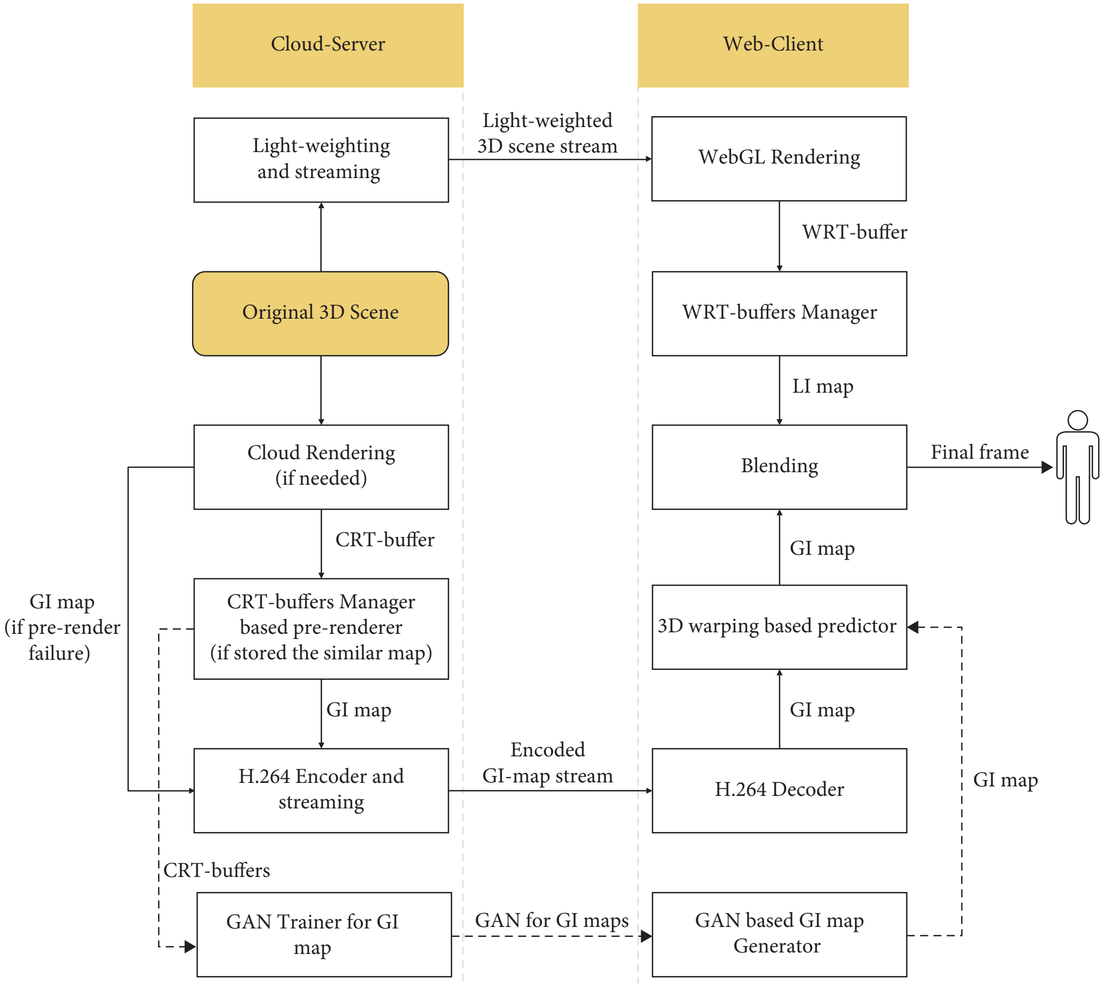

Cloud-Server
Web-Client
Light-weighting and streaming
Light-weighted 3D scene stream
WebGL Rendering
WRT-buffer
Original 3D Scene
WRT-buffers Manager
LI map
Cloud Rendering (if needed)
Blending
Final frame
CRT-buffer
GI map
GI map (if pre-render failure)
CRT-buffers Manager based pre-renderer (if stored the similar map)
3D warping based predictor
GI map
GI map
Encoded GI-map stream
H.264 Encoder and streaming
H.264 Decoder
GI map
CRT-buffers
GAN Trainer for GI map
GAN for GI maps
GAN based GI map Generator

Figure 1: Intelligent Cloud Baking system architecture. Compared with the CB system, our system still contains two modules: cloud rendering and web front-end rendering, but the difference is that we propose a pre-rendering module based on GAN. The new module transfers the pre-rendering task from the CRT-buffers manager to GAN, which solves the problem of cloud storage space limitation and makes full use of the image data stored in CRT-buffers for training GAN.

Table 1: CRT-buffer struct.

Attribute names	Category	Data type
Direct lighting map	L&S-images	Texture
Albedo map		Texture
GI map		Texture
Normal map	G-images	Texture
Depth map		Texture
Direction	Camera information	Vector
Position		Vector

Table 2: WRT-buffer struct.

Attribute names	Category	Data type
Direct lighting map	L&S-images	Texture
Albedo map		Texture
Normal map	G-images	Texture
Depth map		Texture

of eliminating hole artifacts, the Prefetch method, supplements missing pixels by prefetching the new GI map generated by the predicted viewpoint.

In the field of Web3D technology, the camera simulates the human eyes and becomes the primary source of interactive data in the Web3D system. Our Prefetch method predicts the view frustum of the reference camera on the cloud-server side after a change in TIL (TIL is the length of interactive latency in our system). Assuming the network environment is stable, TIL is relatively fixed. We take six general directions, including forwarding, backward, left, right, up, and down, as the basic predicted camera movement directions. Within the TIL, our system calculates the camera view frustums after the movements. Then we set up a new camera whose frustum includes these view frustums, as shown in Figure 3.

First, after the web front-end camera is shifted, the cloud camera needs to include this range, and the cloud camera's view angle θ_t at any time t is shown as follows:

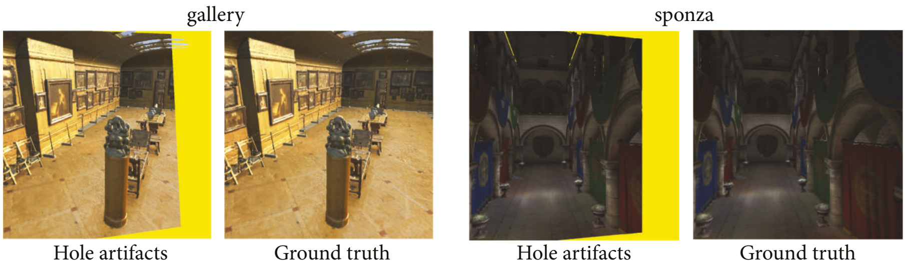


Figure 2: GI map with hole artifact VS. The ground truth. The image processed by 3D Warping has many hole artifacts due to the lack of pixel information, which is obviously different from the ground truth.

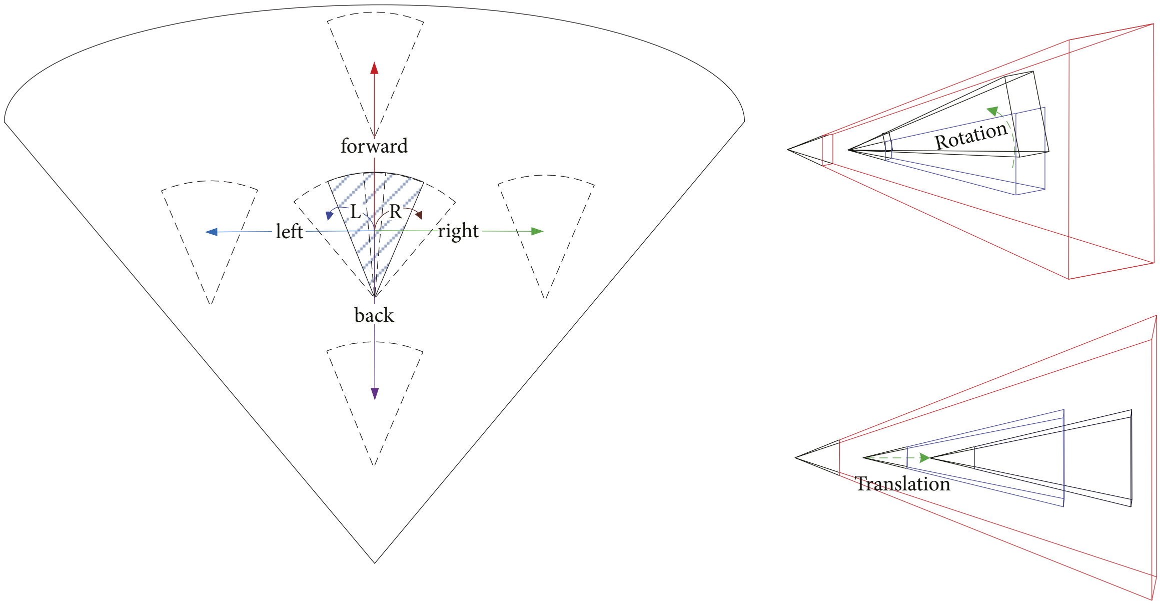


Figure 3: Optimization strategy for 3D warping. This strategy uses the image of the rendered frame to infer the motion vector of the next frame according to the corresponding position of the object in the scene in the pixel and predicts it according to the four directions of movement and the two directions of rotation in the motion vector, which can effectively improve the accuracy of prediction.

$$\theta_t = 2\pi \cot\left(\frac{\max\left\{\left|T_{\mathrm{IL}} * V_t.x\right|, \left|asp * T_{IL} * V_t.y\right|\right\}}{V_t}.z\right) + \alpha. \quad (1)$$

Vt (x, y, z) refers to the moving speed vector of the front-end viewpoint at any time t, asp refers to the aspect ratio of the cross-sectional view of the cloud camera. This formula first compares the moving distance's influence in the horizontal direction (x-direction) and the vertical direction (y-direction) on the view angle, then gets the most significant value and converts it into such an angle. In the same way, after the web front-end camera is rotated, the new view angle of the cloud camera is shown as follows:

$$\theta_r = \max\left\{\alpha + T_{\mathrm{IL}} * V_r.x,\ \arctan\left(\tan\left(T_{\mathrm{IL}} * V_r.y + \alpha\right) * asp\right)\right\}. \quad (2)$$

Vr (x, y, z) is the Web front-end viewpoint's rotation speed. However, under the influence of translation and rotation, the new view angle generated by the camera view volume at the back end of the cloud does not need to be superimposed simultaneously. It is only necessary to use the largest of the two as the new viewing volume angle. The maximum cannot exceed π, as shown in the following formula:

$$\theta = \min\left\{\pi,\ \max\left\{\theta_t, \theta_r\right\}\right\}. \quad (3)$$

The orientation of the viewpoint remains the same, and the position of the viewpoint moves in the opposite direction of the orientation, mainly to place the cloud viewpoint behind the front viewpoint, as shown in the following formula:

$$\begin{cases} P_i.x = P_o.x, \\ P_i.y = P_o.y, \\ P_i.z = P_o.z + V_t.z. \end{cases} \quad (4)$$

Pi (x, y, z) refers to the position of the cloud viewpoint, and Po (x, y, z) refers to the position of the front-end viewpoint.

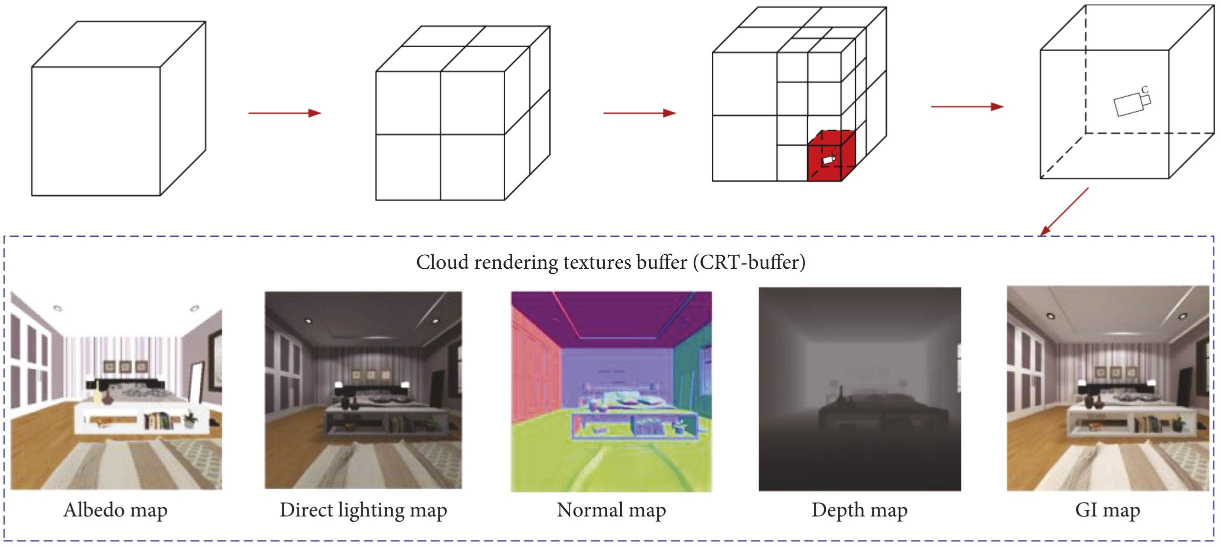


Figure 4: Octree-based CRT-buffers manager. This management method has a positive effect on data access in large-scale scene datasets, using the octagonal tree structure to manage the data in the entire scene, divide the scene data, and store the CRT-buffer data (Albedo maps, Direct lighting map, Normal map, Depth map, and GI map) in the scene to each leaf node, which can effectively improve the system operation efficiency.

5. GAN-based Pre-renderer

For GAN, the management of large-scale data sets for training is very critical. Therefore, we built the CRT-buffer manager, which is an octree-based cloud server data manager, as shown in Figure 4. The nodes of this tree contain a set of viewpoint information in the rendered scene and the image information rendered on these viewpoints, as shown in Table 3. The depth of our octree is determined by the scale of the scene, and we store information in leaf nodes.

Table 3: Node struct of octree in CRT-buffer manager.

Attribute names	Category	Data type
CRT-buffers	L&S-images and camera information	CRT-buffer
Adjacent		CRT-buffer
Position	Node information	Vector
Index		Int

Before the construction of GAN, the pre-rendering task mainly relied on the CRT-buffer manager, which was realized on the assumption that the light source information in the rendered scene did not change. The pre-rendering steps are as follows: (1) Under the assumption that the conditions are established, the system checks whether there is a leaf node in the octree of the manager through the position of the camera. (2) If the leaf node does not exist, our system will directly request the cloud to render the GI map. (3) Otherwise, the system will compare the current camera information with all the camera information in the leaf node one by one. (4) During the comparison process, if it is found that there is a camera "similar" to the current camera's position and orientation in the leaf node, then directly take out the GI map corresponding to the camera's position and send it to the Web front-end. Otherwise, the system will still request the cloud to render the GI map. If the dot product of the positions and directions in the two cameras are all below threshold α ($\alpha < 0.1$), we judge that the two cameras are "similar".

With the accumulation of data in the CRT-buffer manager, the limitation of cloud storage space has become a bottleneck problem for pre-rendering based on the CRT-buffer manager. Therefore, we built a GAN-based Pre-Renderer mechanism, which uses the image generation capabilities of GAN to pre-render, instead of pre-rendering based on the CRT-buffer manager. Meanwhile, a large number of images stored in the CRT-buffer manager provide input data for the construction of GAN.

Our GAN-based Pre-Renderer is derived from our proposed GIGAN system for the rendering of human organs [25]. As shown in Figure 5, the GAN-based Pre-Renderer consists of a Web client and a cloud server and employs WebSocket for network communication. The cloud server trains a conditional generative adversarial network (GAN) to generate a global illumination map (GI map) using a CRT-buffer rendered by the cloud renderer. After training the network, our pre-rendering system sends the generated model to the Web client and then uses it to generate GI maps on the Web client in real-time. Like the GIGAN system, the GAN generator network in our pre-rendering system uses the traditional U-net-based encoder-decoder framework [26]. The discriminator network uses the Markovian patch GAN structure [20]. Their input data is the image dataset in our CRT-buffer, and we prove the validity of these data in the subsequent chapter. In addition, multipath TCP (MPTCP) is considered to be the most potential transmission mechanism to meet the specific requirements of

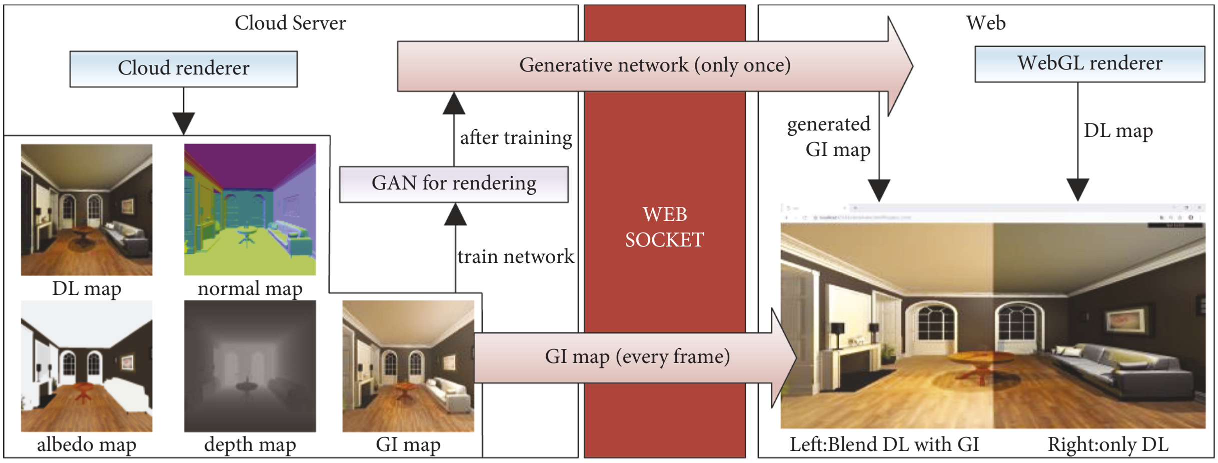


Figure 5: GAN-based Pre-Renderer mechanism. The cloud server transmits the generative model to the web client only once, which changes the previous model in which the cloud server needs to render the GI map and transmit it to the web client every frame. Then the web client can generate the GI map locally in real-time.

multimedia transmission in a multi-homed wireless network environment, which is the main reference for the future transmission mechanism of our system [27].

However, the scene rendered by our system is universal, and the complexity and scale of the 3D models in the scene exceed that of the human organ models rendered in the GIGAN system. Therefore, we optimized GAN to improve the quality of generated images and shorten the training time. The loss function of our GAN is shown as follows:

$$L = L_a + L_c + L_p. \tag{5}$$

Similar to the loss function of GIGAN, we employ the adversarial loss function (La) based on the conditional Wasserstein GAN [28] with gradient penalty to measure the basic information's difference between generated image and ground truth in adversarial processing, as shown in the following formula:

$$\begin{aligned} L_a(G,D) = {} & E_{c,y\ p_{\text{data}}(dg,\ y)}[D(c,y)] \\ & - E_{c\, P_{\text{data}}(c),\ z\ P_z(z)}[D(c,G(z))] \\ & + \lambda_{\text{GP}} E_{c\ P_{\text{data}}(dg),\overline{y}\ p_{\text{GP}}}. \end{aligned} \tag{6}$$

In this formula, we refer to the algorithm of WGAN-GP. G is a generator, D is a discriminator, z~pz (z) is a random noise from a certain distribution (such as normal distribution and uniform distribution), c, y~pdata (c, y) are images, respectively, from the source domain and the corresponding target domain, $\overline{y}$~pgp represents the distribution after linear interpolation between the real data distribution and the generated data distribution. λ GP is a hyperparameter. K represents the expected close value of the gradient during training, and $k = 1$ here.

And we use the contention loss (Lc) to measure the difference of each pixel between the generated image and the ground truth (for details, please refer to the literature [25]), as shown in the following formula:

$$L_c(G) = E_{c,y\sim p_{\text{data}}(c,y)}\left[\|y - G(c,z)\|_1\right]. \tag{7}$$

In addition, we added a new perceptual loss function Lp to the original loss function. We employ the perceptual loss (Lp) to measure the contextual and structural information between generated image and ground truth and apply a pretrained VGG19 network [29] to achieve this. The perceptual loss can be formulated as follows:

$$L_P(y,\widehat{y}) = \frac{1}{C_i H_i W_i} E_{y\sim p_{data},\widehat{y}\sim p_{gp}}\left[\|\theta_i(y) - \theta_i(\widehat{y})\|_2\right]. \tag{8}$$

In formula 8, C_i, $H_{i,}$ and $W_{i,}$ respectively, represent the channel number, width and height of the image features. $\theta_i(y)$ indicates the i-th layer of the VGG19 network (after activation). The new loss function plays an effective role in ensuring the quality of the output GI map.

As shown in Figure 6, the image data set in the CRT-buffer is collected as the input of the generator with skip connections and then passes 5 downsampling layers and 5 upsampling layers before generating the GI map. LeakyReLu is used as the activation function in the entire downsampling process, while ReLu and tanh (the last layer only) are used as the activation function in the upsampling process. The discriminator is composed of 4 encoders and uses LeakyReLU as the activation function. During training, the network randomly reads data from the data set in batch size to 4, and G-D alternately uses mini-batch stochastic gradient descent and Adam optimizer with learning rate = 0.0001 to update the weight of the network. Compared with GIGAN, the GAN training time of our pre-rendering system is shortened by 20%–30%.

6. Experimental Results and Analysis

The test environment of our system is as follows: Our cloud server is equipped with two Intel Xeon Silver 4114 2.2 GHz CPUs, one Nvidia Quadro P5000 GPU, and 128 GB of RAM, and the server is running Windows Server 2012. For the web client, we use a laptop with an Intel Core i7-7700HQ 2.8 GHz CPU, an Nvidia GeForce GTX1060 M GPU, and

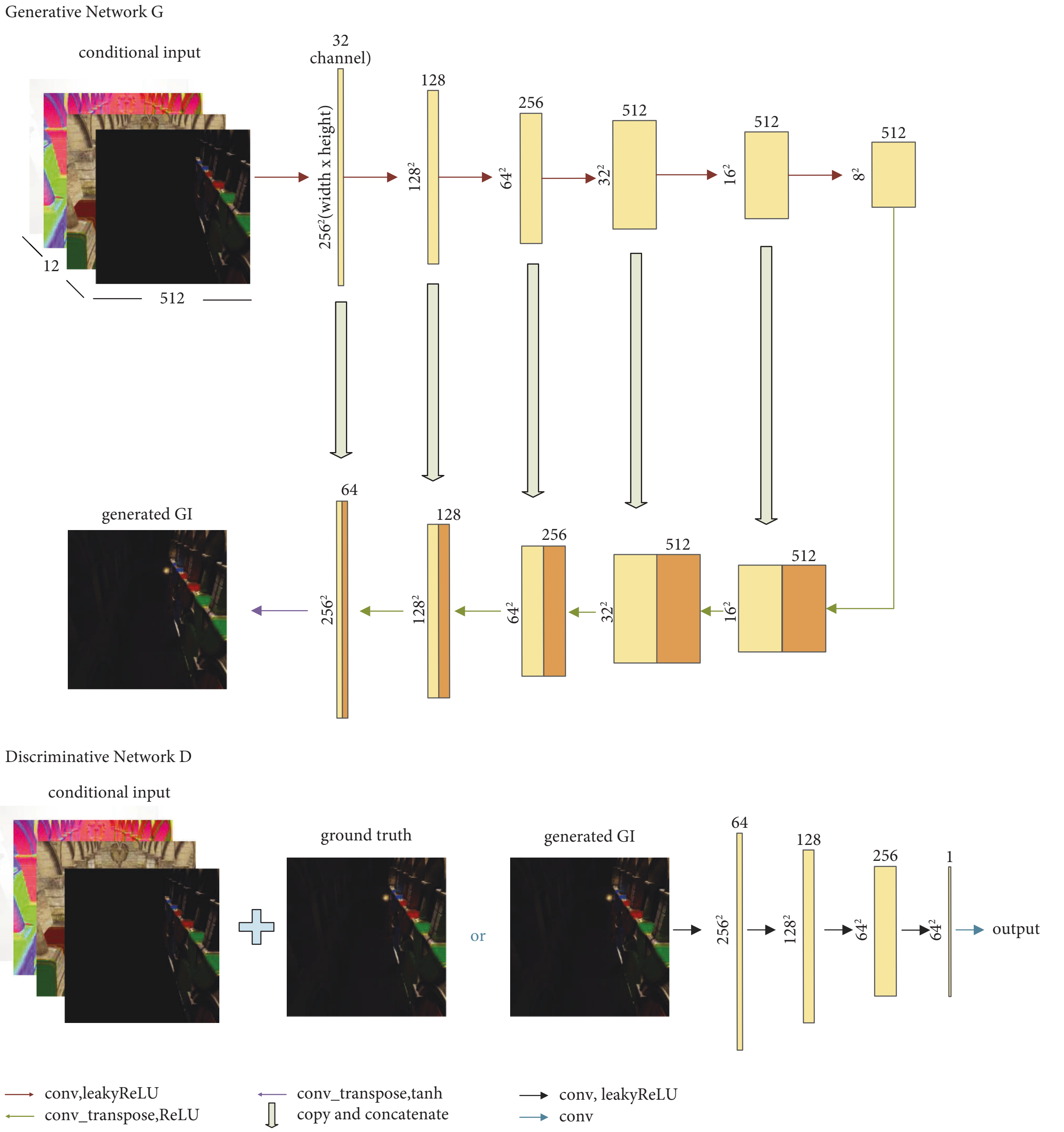


Figure 6: Overall conditional generative adversarial network architecture. The generator network adopts a 5-layer U-NET structure, in which the convolution and deconvolution operations can be regarded as the process of encoding and decoding. And the discriminant network adopts a 4-layer full convolutional network.

8 GB of RAM. The laptop runs Windows 10 and uses Google Chrome version 71 as a web browser.

For verifying the rationality of the input of our GAN and the effectiveness of this system, we conducted the following test: (1) First, we enumerate a combination of multiple types of image data as the input data of the generative confrontation network. (2) In the cloud back-end, we generate various GANs based on these different sets of input data. (3) We pass these GANs to the Web front-end and generate new GI maps based on them on this end. (4) Finally, we test the image quality of these GI maps generated by different GANs and compare them.

Direct lighting information is part of global illumination, and the albedo is directly involved in the calculation of global illumination. Therefore, we use the DL map and albedo map data storing these two information as the most

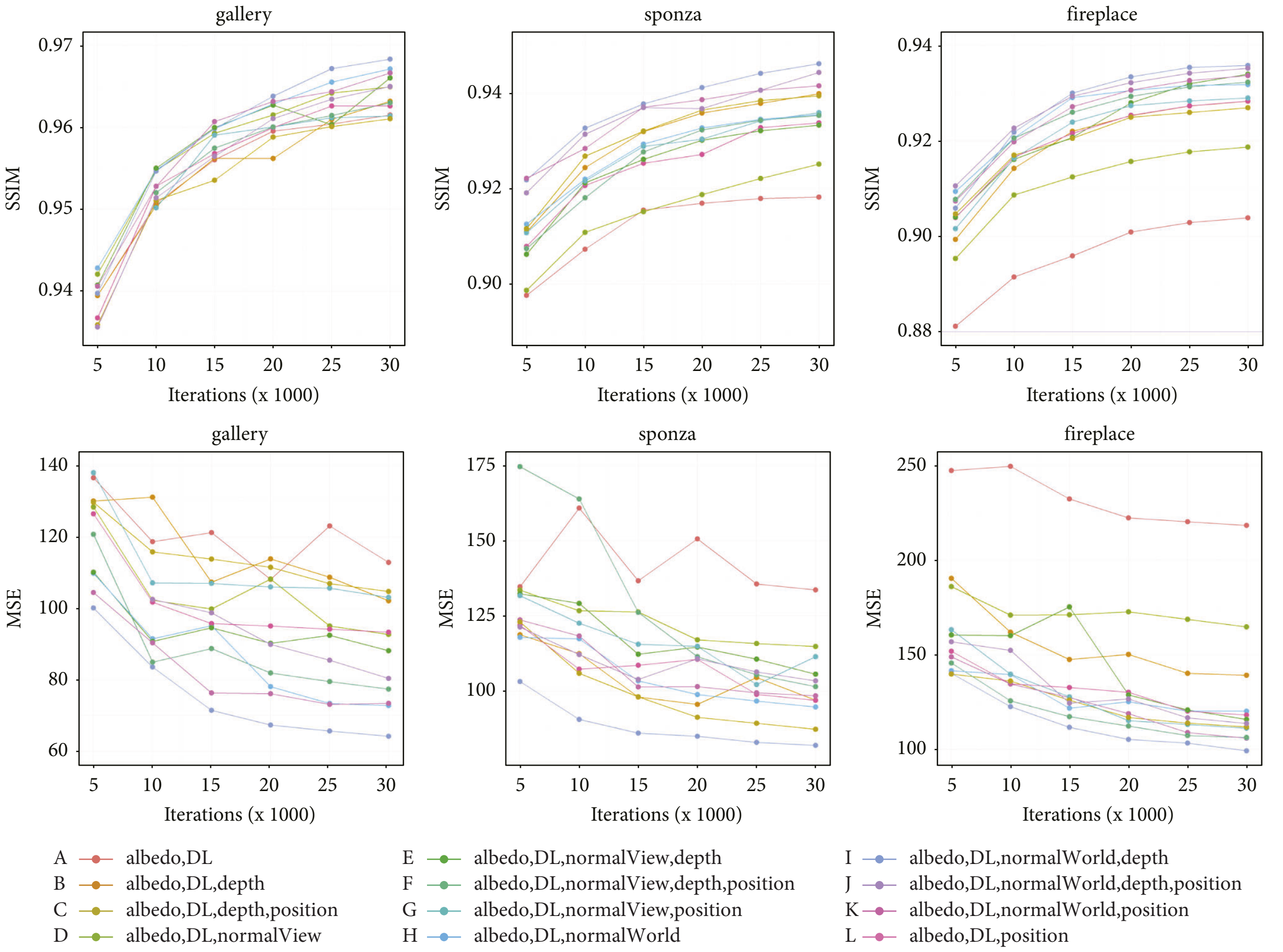


Figure 7: Network iterations VS. SSIM and MSE in different input configurations (test scene is sponza). We can compare the difference between GI map generated by different GANs and Ground truth under the same number of iterations.

important GAN input data to generate the final GI map. In addition, we also selected position map, depth map, normalWorld map (normal map in world space), and noramlView map (normal map in view space) from the G-buffers data. We take the random combination of the pictures elected in G-buffers and the previous two pictures as GAN's input configuration, as shown in the dotted box in Figure 7. In the end, we get multiple sets of generative adversarial network models.

We judge the pros and cons of our GANs' model by testing the quality of the final generated GI map. This paper uses structural similarity (SSIM) and mean square error (MSE) to evaluate the similarity between the generated image and the real image. The former measures the similarity by comparing the brightness, contrast, and structure of the two images, while the latter measures it from the error between the corresponding pixels of the two images. Zinner et al. proposed that the image quality is acceptable when SSIM is greater than 0.88 [22], so our paper uses 0.88 as the image quality threshold and sets both SSIM and MSE to be calculated in the image's RGB color space (range 0–255). In addition, we make the cameras in the test set move along a different path from the training set to test the generalization of the model. In Figure 7, different colors represent different input configurations. The horizontal axis represents the number of iterations of the network, and the vertical axis represents the SSIM (higher is better) or the MSE (lower is better) between the generated image and the real image.

As shown in Figure 7, most GANs obtained after a small number of training iterations can make the generated image's quality exceed the basic threshold. We judge the direct illumination information occupies a higher ratio in the global illumination image, which enables GAN to quickly fit and generate a high-quality GI map, and all our input configurations include this information. After 20,000 training iterations, the changes of SSIM and MSE between the real image and the image generated by most GANs have stabilized. Therefore, we consider 20,000 times is the ideal threshold for our GAN training times.

In addition, based on the viewpoint correlation of our system, we use the normal map of the view space as the input data of GAN. But the experimental results prove that using the normal map of world space as the input data of GAN can make the final generated image quality higher, which means

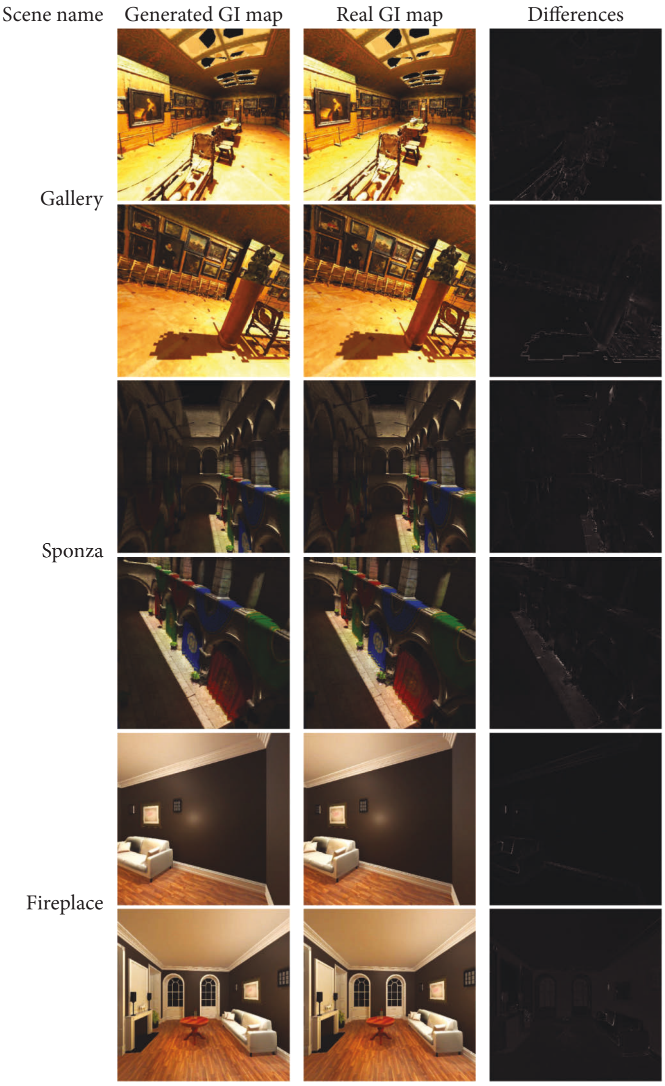


Figure 8: Generated GI map VS. Real GI map. The difference between the differential GI map and the Real GI map pixel-by-pixel difference operation can be more intuitively recognized. The darker the color, the smaller the difference between the corresponding pixels These results show that the generated GI maps are very close to the real GI maps.

that the normal map of world space has more effective information than the normal map of view space. Note that the data between the three pairs of different input configurations in Figure 7 illustrate the above results, including *E* and *H*, *F* and *I*, *G* and *J*.

Finally, we found that among all input configurations, the generated image obtained by the GAN obtained by the I^{th} input configuration (albedo, DL, normal, and depth) has the highest SSIM and the lowest MSE value, and the entire training process is relatively stable. Therefore, we use the I-th input configuration, the final results are shown in Figure 8. These results show that the GI maps generated by this GAN are very close to the real GI maps.

7. Conclusion

This paper attempts to combine cloud rendering technology with artificial intelligence technology. We propose a complete architecture of a smart cloud rendering system for Web3D based on the CloudBaking system and GAN. Our system uses a trained neural network to generate rendered images and eventually partially replaces the hardware's rendering capabilities. Experimental results prove that the GAN-based intelligent rendering system for Web3D can complete rendering tasks while saving and protecting rendering resources.

Although the use of GAN effectively improves the speed of real-time rendering, neural network cannot completely solve the inevitable delay. In order to reduce latency, the architectural optimization of cloud rendering systems is usually considered. At present, 5G network and edge computing technology are fully utilized to optimize the architecture of cloud rendering system, which reduces the interaction delay of the system. In addition, the training process relies on the accumulation of a large number of prerendered images, which is also a challenge to storage space.

And the use of different neural network structures has a crucial impact on the result. These are the areas that can be improved in our future work.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was part of the research supported by the Science and Technology Program of Educational Commission of Jiangxi Province, China (DA202104172), the Innovation and Entrepreneurship Course Program of Nanchang Hangkong University (KCPY1910), and the Teaching Reform Research Program of Nanchang Hangkong University (JY21040).

References

[1] C. Liu, J. Jia, Q. Zhang, and L. Zhao, "Lightweight websim rendering framework based on cloud-baking," in *Proceedings of the 2017 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*, pp. 221–229, Singapore, May 2017.

[2] C. Liu, W. T. Ooi, J. Jia, and L. Zhao, "Cloud baking: collaborative scene illumination for dynamic Web3D scenes," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 14, no. 3, pp. 1–20, 2018.

[3] L. Chittaro and R. Ranon, "Web3D technologies in learning, education and training: Web3D technologies in learning, education and training: Motivations, issues, opportunitiesotivations, issues, opportunities," *Computers & Education*, vol. 49, no. 1, pp. 3–18, 2007.

[4] A. Evans, M. Romeo, A. Bahrehmand, J. Agenjo, and J. Blat, "3D graphics on the web: 3D graphics on the web: A survey survey," *Computers & Graphics*, vol. 41, pp. 43–61, 2014.

[5] C. Marion and J. Jomier, "Real-time collaborative scientific WebGL visualization with WebSocket," in *Proceedings of the 17th International Conference on 3D Web Technology*, pp. 47–50, California, CL, USA, July 2012.

[6] H. Jacinto, R. Kéchichian, M. Desvignes, R. Prost, and S. Valette, "A web interface for 3D visualization and interactive segmentation of medical images," in *Proceedings of the 17th International Conference on 3D Web Technology*, pp. 51–58, California, CL, USA, Augest 2012.

[7] H. Rahaman and B. K. Tan, "Interpreting digital heritage: a conceptual model with end-users' perspective," *International Journal of Architectural Computing*, vol. 9, no. 1, pp. 99–113, 2011.

[8] X. Liu, N. Xie, K. Tang, and J. Jia, "Lightweighting for Web3D visualization of large-scale BIM scenes in real-time," *Graphical Models*, vol. 88, pp. 40–56, 2016.

[9] T. Mzoughi, S. D. Herring, J. T. Foley, M. J. Morris, and P. J. Gilbert, "WebTOP: a 3D interactive system for teaching and learning optics," *Computers & Education*, vol. 49, no. 1, pp. 110–129, 2007.

[10] Y. Li, K. Brodlie, and N. Phillips, "Web-based VR training simulator for percutaneous rhizotomy," in *Medicine Meets Virtual Reality 2000*, pp. 175–181, IOS Press, Amsterdam, Netherlands, 2000.

[11] V. Ramasundaram, S. Grunwald, A. Mangeot, N. B. Comerford, and C. Bliss, "Development of an environmental virtual field laboratory," *Computers & Education*, vol. 45, no. 1, pp. 21–34, 2005.

[12] C. Liu, J. Jia, Y. Ge, and N. Xie, "Web3D online virtual education platform for touring huangyangjie battlefield scenario over internet," in *Proceedings of the International Conference on Technologies for E-Learning and Digital Entertainment*, pp. 63–76, Springer, 2016.

[13] D. Koller, M. Turitzin, M. Levoy et al., "Protected interactive 3D graphics via remote rendering," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 695–703, 2004.

[14] C. Crassin, D. Luebke, M. Mara et al., "CloudLight: a system for amortizing indirect lighting in real-time rendering," *Journal of Computer Graphics Techniques*, vol. 4, no. 4, pp. 1–27, Hangzhou, China, April 2015.

[15] K. Bugeja, K. Debattista, and S. Spina, "An asynchronous method for cloud-based rendering," *The Visual Computer*, vol. 35, no. 12, pp. 1827–1840, 2019.

[16] J. H. Mueller, P. Voglreiter, M. Dokter et al., "Shading atlas streaming," *ACM Transactions on Graphics*, vol. 37, no. 6, pp. 1–16, 2018.

[17] I. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[18] P. Isola, J. Y. Zhu, T. Zhou, and A. Efros, "A Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1125–1134, Honolulu, HI, USA, July 2017.

[19] Z. Yi, H. Zhang, P. Tan, and M. Gong, "Dualgan: unsupervised dual learning for image-to-image translation," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2849–2857, Venice, Italy, October 2017.

[20] C. Li and M. Wand, "Precomputed real-time texture synthesis with Markovian generative adversarial networks," in *Proceedings of the European Conference on Computer Vision*, pp. 702–716, Springer, Amsterdam, The Netherlands, October 2016.

[21] C. Ledig, L. Theis, F. Huszár et al., "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4681–4690, Honolulu, HI, USA, July 2017.

[22] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[23] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y. G. Jiang, "Pixel2mesh: generating 3d mesh models from single rgb images," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 52–67, Munich, Germany, September 2018.

[24] L. Wen, J. Jia, and S. Liang, "LPM: lightweight progressive meshes towards smooth transmission of Web3D media over internet," in *Proceedings of the 13th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its*

Applications in Industry, pp. 95–103, Shenzhen China, November 2014.

[25] N. Xie, Y. Lu, and C. Liu, “Web3D client-enhanced global illumination via GAN for health visualization,” *IEEE Access*, vol. 8, Article ID 13281, 2019.

[26] O. Ronneberger, P. Fischer, and T. Brox, “U-net: convolutional networks for biomedical image segmentation,” in *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, Springer, Munich, Germany, October 2015.

[27] Y. Cao, L. Zeng, Q. Liu, G. Lei, M. Huang, and H. Wang, “Receiver-assisted partial-reliable multimedia multipathing over multi-homed wireless networks,” *IEEE Access*, vol. 7, Article ID 177689, 2019.

[28] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in *Proceedings of the International Conference on Machine Learning*, pp. 214–223, Sydney, Australia, August 2017.

[29] C. Ledig, L. Theis, F. Huszár et al., “Photo-realistic single image super-resolution using a generative adversarial network,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4681–4690, Honolulu, HI, USA, July 2017.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 1812273, 14 pages
https://doi.org/10.1155/2022/1812273

Research Article

Attention-Based LSTM Model for IFA Detection in Named Data Networking

Xin Zhang,[1,2] **Ru Li**,[1,2] **and Wenhan Hou**[1,2]

[1]*Inner Mongolia Key Laboratory of Wireless Networking and Mobile Computing, Hohhot 010021, China*
[2]*College of Computer Science, Inner Mongolia University, Hohhot 010021, China*

Correspondence should be addressed to Ru Li; csliru@imu.edu.cn

Received 28 April 2022; Revised 21 May 2022; Accepted 28 May 2022; Published 21 July 2022

Academic Editor: Kai Wang

As one of the next generation networks, Named Data Networking (NDN) performs well on content distribution. However, it is vulnerable against a new type of denial-of-service (DoS) attacks, interest flooding attacks (IFAs), one of the fatal threats to NDN. The attackers request nonexist content to occupy the Pending Interest Table (PIT), and it causes the degradation of network performance. Because of the great harm and strong concealment of this attack, it is urgent to detect and throttle the attack. This paper proposes a detection mechanism based on Long Short-Term Memory (LSTM) with attention mechanism, which uses sequence with different treatments. Once IFA is detected, the Hellinger distance is used to recognize malicious Interest prefix. The simulation results show that the proposed scheme can resist IFA effectively compared to state-of-the-art schemes.

1. Introduction

The purpose of traditional network architecture based on TCP/IP is to meet the end-to-end data transmission, which cannot meet the diversified needs today. Therefore, the researchers began to study new network architectures. Information Centric Networking (ICN) [1] aims to build a new content-centric future network architecture, and it transforms the current host-centric communication mode into the content-centric network communication mode. Typical representative projects of ICN include information-oriented network architecture (Network of Information, NetInf) [2], publish/subscribe Internet routing paradigm, and publish/subscribe Internet topology (PSIRP/PURSUIT) [3], Data-Oriented Network Architecture (DONA) [4], Content Centric Networking (CCN) [5], and Named Data Networking (NDN) [6]. The most representative ICN architecture is NDN, which was proposed by Zhang Lixia of UCLA (University of California-Los Angeles) and Van Jacobson of Xerox PARC (Xerox Palo Alto Research Center) in 2010. The architecture of NDN is shown in Figure 1.

In the NDN network, there are two types of packets: Interest packet and Data packet [6]. The users send Interest packet to request content, and the returned content is called Data packet. There are three data structures in NDN: content store (CS), Pending Interest Table (PIT), and forwarding information base (FIB) [6]. NDN implements routing and forwarding via these three data structures:

(i) FIB: it stores the interface information pointing to the specified content, and the Interest packet is forwarded according to the FIB.
(ii) PIT: it records the unsatisfied Interest packet and the corresponding interfaces and can aggregate the Interest packets, and the Data packets are returned in the original way according to the interface information of the PIT.
(iii) CS: the router caches the received Data packet to realize intranetwork caching and reduces the delay for users to obtain data.

The NDN forwarding process of Interest packet and Data packet is shown in Figure 2.

When an NDN router receives an Interest packet, first it checks if CS has a matching data. If so, the router returns the Data Packet. Otherwise, the router checks whether PIT has a matching entry. If it exists, the router adds incoming

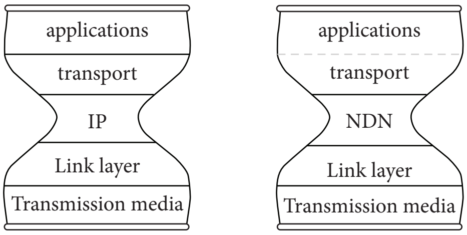


Figure 1: TCP/IP architecture vs NDN architecture [6, 7].

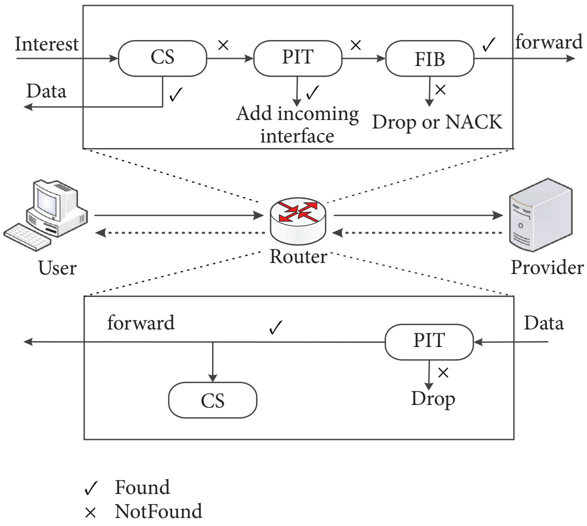


Figure 2: NDN forwarding process [6].

interface of the Interest packet to the entry. Otherwise, the router forwards Interest packet based on the FIB. When receiving a Data packet, the router first checks if PIT has a matching entry. If it exists, the router returns the Data packet based on the information of the PIT and caches the Data Packet. Otherwise, the router will drop the Data packet.

Denial of service (DoS) and distributed denial of service (DDoS) are rampant in the traditional TCP/IP architecture [8]. NDN can mitigate the impact of DDoS in TCP/IP architecture. However, the researchers discover a new type of DDoS attack called IFA [8]. As shown in Figure 3, the attacker forges a number of fake Interest packets to consume the memory resources of routers, which cause the degradation of network performance.

The IFA attack has great harm and strong concealment, and the researchers have tried various defend mechanisms, mainly including machine learning and statistical method. Due to the characteristics of network traffic, it is difficult to accurately identify attacks of a single time interval, resulting in low accuracy of attack detection. This paper uses past data through sliding window and proposes an attention-based Long Short-Term Memory (LSTM) [9] for IFA detection. Once IFA is detected, the Hellinger distance [10] is used to identify the malicious prefix.

The contributions of this paper are summarized as follows:

(1) This paper uses the LSTM model with attention mechanism to detect IFA by exploiting the past data sequence and with different treatments

(2) This paper proposes a Hellinger distance-based malicious Interest prefix identify mechanism

(3) The simulation results show that the scheme proposed can detect IFA effectively

The rest of the paper is organized as follows: Section 2 gives a review of related works. Section 3 presents detection mechanism and mitigation mechanism in detail. Section 4 gives an evaluation of the proposed mechanism and compares the proposed mechanism with state-of-the-art mechanism. Finally, Section 5 concludes the paper.

2. Related Works

Various literature works have been proposed on detecting and mitigating the IFA. Some approaches use machine learning to detect IFA. In paper [11], linear SVM and SVM with Gaussian radial basis kernel function were used to detect IFA. It consisted of two phases: the training phase and the test phase. In paper [12], the Isolation Forest was used to calculate the abnormal score of each Interest prefix at the end of each fixed time interval to detect abnormal Interest packet prefix. In paper [13], the deep reinforcement learning was used to detect IFA. In paper [14], the naïve Bayes (NB), J48 decision tree, multilayer perceptron with backpropagation (BP), and radial basis function (RBF) network were used to detect IFA. In paper [15], the authors used multilayer perceptron (MLP) with backpropagation (BP), radial basis function (RBF) network with particle swarm optimization (PSO), JAYA and teaching–learning-based optimization (TLBO), linear support vector machine (SVM), and fine k-nearest neighbours (KNN) to detect the attack. In paper [16], the authors used association rule algorithm to find the correlation between features and used decision tree algorithm to detect the attack.

Some approaches use the mathematical model to detect IFA. In paper [17], every NDN router computed the Gini impurity to detect IFA by measuring the Interest name in a router. In paper [18], the Theil index was used to detect IFA and the Interest packets were divided into groups by Theil entropy to evaluate the intragroup and intergroup difference of Interest name distribution. In paper [19], two traffic features were used to establish confidence interval, respectively, to detect IFA. In paper [20], the authors used mean and variance of packet hop counts to distinguish legitimate users from malicious users. In paper [21], the authors used hash-based security label to identify the malicious prefix. In paper [22], the authors used wavelet analysis to detect IFA. In paper [23], the routers used active queue management (AQM) to defend IFA. In paper [24], each edge router used token-based router monitoring policy (TRM) to mitigate the IFA by controlling the data requestors. The detection method used in the related work is shown in Table 1. The main drawback of existing IFA detection method is counting the traffic information on a fixed time interval, which ignores the temporal relationship of traffic.

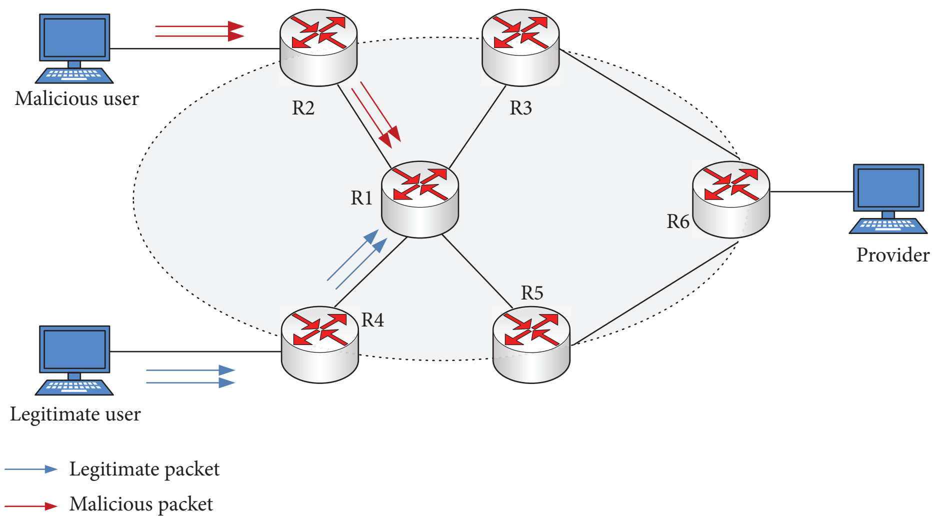


Figure 3: IFA sample.

Table 1: Comparison between related paper.

Paper	Year	Offline	Online	Detection method
[12]	2021	✗	✓	Isolation forest
[16]	2021	✓	✗	Association rules + decision tree
[23]	2021	✗	✓	AQM
[24]	2021	✗	✓	Token
[19]	2020	✗	✓	Confidence interval
[21]	2020	✗	✓	Hash
[11]	2019	✓	✓	SVM
[18]	2019	✗	✓	Theil index
[25]	2019	✗	✓	Hypothesis testing
[26]	2019	✗	✓	AQM
[13]	2020	✗	✓	Deep reinforcement learning
[17]	2018	✗	✓	Gini impurity
[14]	2019	✓	✗	MLP with BP RBF classifier J48 Naive Bayes
[20]	2018	✗	✓	Mean-variance
[15]	2017	✓	✗	MLP with BP RBF with PSO RBF with JAYA RBF with TLBO SVM linear Fine KNN
[22]	2017	✗	✓	Wavelet analysis

3. Detection Mechanism Based on Attention Mechanism with LSTM

This section gives an overview of proposed defend mechanism, detection mechanism, and mitigation mechanism.

3.1. Overview. The defend mechanism mainly consists of five parts, the data collection module, the data preprocessing module, the detection module, the response module, and the mitigation module, as shown in Figure 4.

In the data collection module, the traffic data is collected and it is then input to the preprocessing module. In the preprocessing module, the traffic characteristics are extracted. The traffic characteristics are used to detect IFA in the detection module. Once IFA is detected, the response module will start identify the malicious prefix. Finally, the mitigation module uses malicious prefix to limit the malicious Interest packet.

3.2. Long Short-Term Memory. Deep learning is popular and is used in various applications. Recurrent neural network (RNN) [27] is a type of deep learning methods, which can be used to detect anomaly. However, there is a gradient vanishing problem in RNN [28]. Long Short-Term Memory (LSTM) [9] is an improved version of RNN, which solves the problem of RNN. The LSTM structure is shown in Figure 5.

It mainly includes three structures, input gate, forget gate, and output gate, which are used to update the LSTM cell as follows [9]:

$$\begin{aligned} f_t &= \sigma\left(W_f\left[h_{t-1}, x_t\right] + b_f\right), \\ i_t &= \sigma\left(W_i\left[h_{t-1}, x_t\right] + b_i\right), \\ \widetilde{C}_t &= \tanh\left(W_C\left[h_{t-1}, x_t\right] + b_C\right), \\ C_t &= f_t * C_{t-1} + i_t * \widetilde{C}_t, \\ o_t &= \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right), \text{ and} \\ h_t &= o_t * \tanh\left(C_t\right), \end{aligned} \tag{1}$$

where W is the weight, b is the bias, h_t is the hidden state at time step t, and x_t is the input at time step t.

3.3. Attention Mechanism. The Attention mechanism is inspired by human attention behaviour and is well applied to deep learning.

In paper [29], the attention mechanism was proposed. Given an input $X = [x_1, x_2, \ldots, x_N] \in R^{D\times N}$, where N is the length of input, $x_n \in R^D$, $n \in [1, N]$, and D is the number of dimensions in each time step, the calculation of the attention mechanism is divided into two steps: first calculate the attention probability of all input and then calculate the

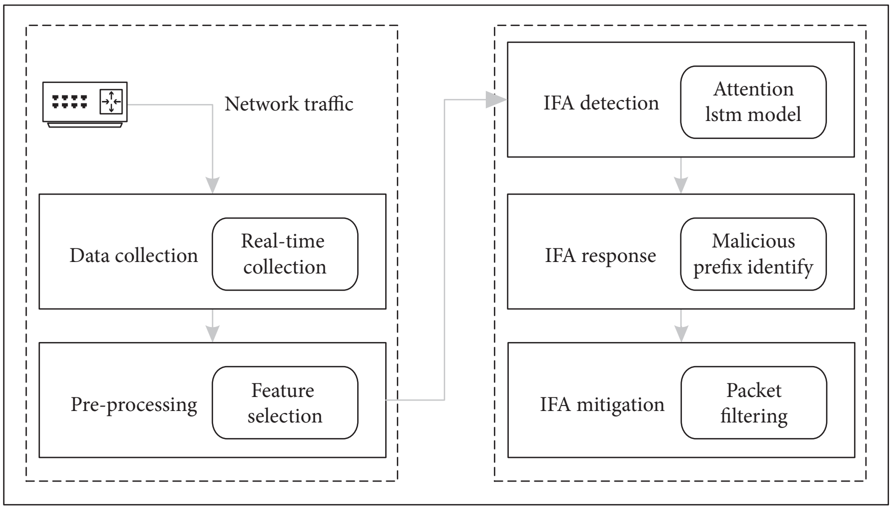


Figure 4: The architecture of defend mechanism.

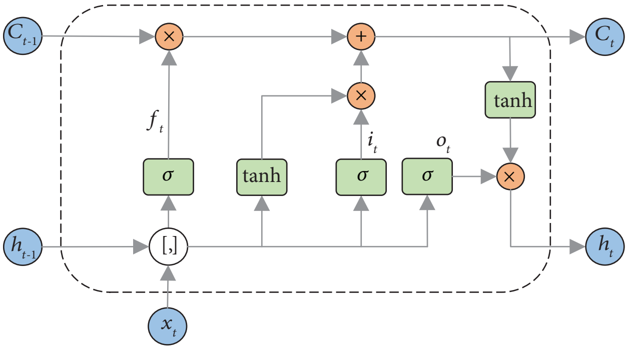


Figure 5: An LSTM cell structure [9].

weighted average of the input information according to the attention probability.

3.4. Detection Mechanism. This section presents the detection mechanism in detail. First, some used notations are listed and some features are defined. The notations used are listed in Table 2.

Definition 1. (Router PIT Utilization Size). It denotes the number of PIT entries in PIT during one time slice.

$$U(t_i, R_j) = e(t_i, R_j). \tag{2}$$

Definition 2. (Router Interest Satisfaction Ratio). It denotes the number of Data packets received to the number of Interest packets received in one time slice.

$$S(t_i, R_j) = \frac{\varphi(\phi(t_i, R_j))}{\phi(t_i, R_j)}. \tag{3}$$

Definition 3. (Router Interest Request Frequency). It denotes the number of Interest packets received in one time slice.

$$I(t_i, R_j) = \phi(t_i, R_j). \tag{4}$$

Definition 4. (Router Data Reply Frequency). It denotes the number of Data packets replied in one time slice.

$$r(t_i, R_j) = \varphi(\phi(t_i, R_j)). \tag{5}$$

The feature calculation is shown in Algorithm 1.

The detection mechanism detects IFA through a sliding window, as shown in Figure 6.

A network traffic formally as a time series: $Z = \{z^1, z^2, \ldots, z^i, \ldots, z^F\}$, which consists of F time steps. $z^i\,(1 \le i \le F)$ represents the i th time step. For each sliding window, which consists of φ time steps, the detection model is used to classify the sliding window as legitimate or malicious.

Figure 7 shows the LSTM with attention mechanism for IFA detection. The attention mechanism can improve the performance of LSTM by discriminatively utilizing each step of hidden state information [30]. Therefore, this paper uses the traditional LSTM with attention mechanism to detect IFA. The hidden states of each step are multiplied with attention weights.

In LSTM layer, the input of each step is mapped to a hidden state.

$$h_i = \mathrm{LSTM}(z_i), \quad i \in [1, F], \tag{6}$$

where h_i is the hidden state at time step i and z_i is the input at time step i.

In attention layer, the hidden state of each step is input to a subsequent attention layer. It takes the form as follows [31]:

$$\begin{gathered}\mathscr{H} = \sum_{t=1}^{N} \alpha_t h(t) \text{ and} \\ \alpha_t = \frac{\exp(g_t(W_t, h(t)))}{\sum_{t=1}^{N} \exp(g_t(w_t, h(t)))},\end{gathered} \tag{7}$$

Table 2: Notation used.

Notation	Description
t_i	The i-th time slice
R_j	The j-th router
$\phi(t_i, R_j)$	The number of receiving Interests of the j-th router in the i-th time slice
$\varphi(\phi(t_i, R_j))$	The number of receiving corresponding Data packets
$e(t_i, R_j)$	The number of PIT entry of router j at the i-th time slice

```
    Input:
    $\varepsilon$ ▷ The time slice size
    Output:
    $i$ ▷ The request frequency
    $r$ ▷ The reply frequency
    $s$ ▷ The satisfaction ratio
(1) procedure IncomingInterest(slice $\varepsilon$)
(2)     $i \longrightarrow i + 1$
(3) end procedure
(4) procedure IncomingData(slice $\varepsilon$)
(5)     $r \longrightarrow r + 1$
(6) end procedure
(7) $s \longrightarrow r/i$
(8) return $i$ $r$ s
```

Algorithm 1: Interest features computing.

```
     Input:
     $\varepsilon$ ▷ The time slice size
     $\varphi$ ▷ The sliding window size
     $Thr$ ▷ Detection threshold
     Output:
     Detection result
 (1) Compute the metrics during time slice $\varepsilon$
 (2) for the consecutive time step with length $\varphi$ do
 (3)     fed the sequence $Z$ to the detection model
 (4)     $y = \text{LSTM}Att(Z)$
 (5)     if $y > Thr$ then
 (6)         return legitimate
 (7)     else
 (8)         return malicious
 (9)     end if
(10) end for
```

Algorithm 2: LSTM with attention mechanism-based detection.

where α_t is the weight for each time step and $g_t(\cdot)$ is a fully connected layer with ReLU activation and parameters W_t.

The illustration of attention mechanism is shown in Figure 8.

In output layer, the attention layer results $\mathscr{H}$ is input to a fully connected layer with sigmoid activation to obtain the final result.

$$\text{output} = \text{simoid}(v). \tag{8}$$

The detection mechanism is shown in Algorithm 2.

The algorithm works as mentioned in the following steps:

Step (1): count the traffic information in time slice ε, use Algorithm 1

Step (2): when the sliding window size is φ, fed to the detection model, get output y

Step (3): if the detection result is legitimate, forward the sliding window and return to Step (2)

Step (4): if the detection result is malicious, trigger the malicious prefix identification mechanism

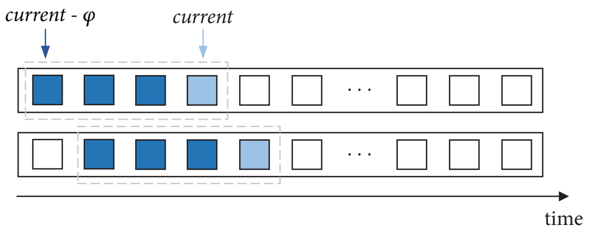


Figure 6: The sliding window.

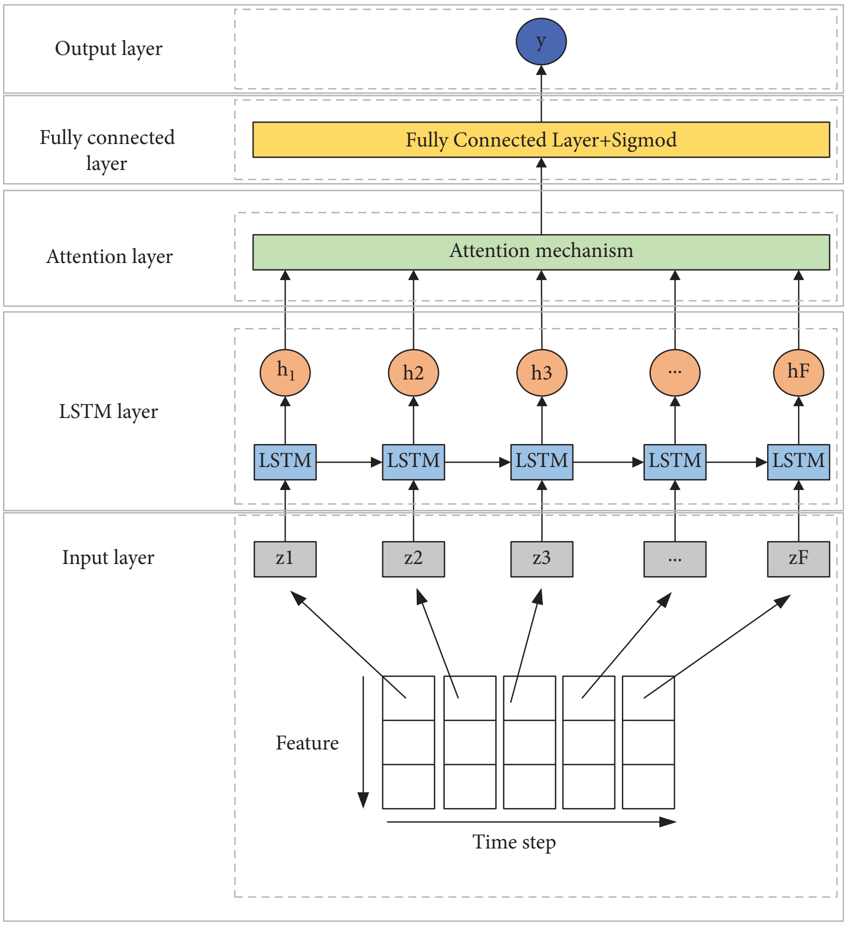


Figure 7: The LSTM with attention mechanism.

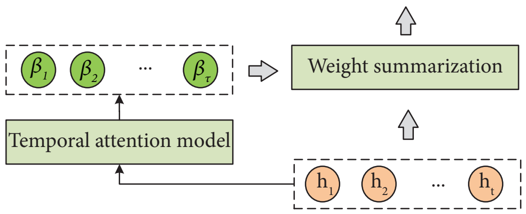


Figure 8: Illustration of temporal attention mechanism.

3.5. Response Mechanism. This paper recognizes the malicious Interest prefixes based on the Hellinger distance [10]. The Hellinger distance is used to measure the deviation between two probability distributions independent of parameters.

$$H(\mathbb{P},\mathbb{Q}) = \frac{1}{\sqrt{2}}\sqrt{\sum_{i=1}^{n}\left(\sqrt{p_i}-\sqrt{q_i}\right)^2}, \quad p_i \ge 0;\ q_i \ge 0, \tag{9}$$

where $\mathbb{P}$ and $\mathbb{Q}$ are two probability distributions, $\mathbb{P}$ and $\mathbb{Q}$ are n-tuples $(p_1, p_2, .., p_n)$, and $(q_1, q_2, .., q_n)$, $\sum_i p_i = 1$, and $\sum_i q_i = 1$.

The malicious prefix recognition process is shown in Algorithm 3.

3.6. Mitigation Mechanism. When malicious prefixes are recognized, the router will send notification packet that

```
Input:
    Interest prefix distribution when IFA is detected: $\mathbb{P}$
    Interest prefix distribution before IFA is detected: $\mathbb{Q}$
    Interest prefix set: $I$
Output:
    Malicious prefix set
$\mathbb{Q}' = \mathbb{Q}$
for prefix$_i \in I$ do
    $\mathbb{Q}'_i = \mathbb{P}_i$
    calculate the Hellinger distance $H(\mathbb{Q}'_i, \mathbb{Q})$
    if $H(\mathbb{Q}'_i, \mathbb{Q}) > thr$ then
        add prefix$_i$ to malicious prefix set
    end if
end for
return malicious prefix set
```

Algorithm 3: Hellinger distance-based malicious prefix recognition.

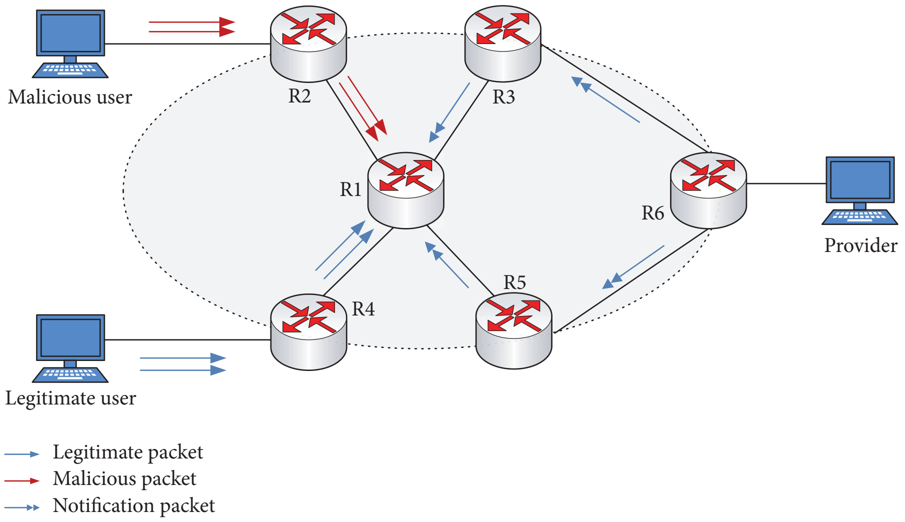


Figure 9: IFA mitigation sample.

includes the malicious prefixes to the downstream router, as shown in Figure 9. The downstream routers extract the malicious prefix and limit its sending rate when receiving the notification packet.

4. Performance Evaluation

In order to evaluate the performance of the proposed scheme, this paper conducts a set of simulations in ndnSIM [32]. Then, this paper compares the proposed scheme with the state-of-the-art defend scheme. The simulations parameters are shown in Table 3.

This paper considers tree topology as shown in Figure 10. The tree topology which is one of the most severely affected by the IFA is widely used in detection mechanism evaluation of IFA.

In tree topology, *Rx* denotes the NDN router, *Cx* denotes the legitimate user, *Px* denotes the data provider, and *Ax* denotes the malicious user. The red lines denote connections between the malicious user and NDN router, the green lines denote connections between the legitimate user and NDN router, the black lines denote connections between NDN routers, and the blue lines denote the connections between the data provider and NDN router.

In tree topology, there are 9 legitimate users and 7 malicious users. The simulation lasts 800s. The legitimate users issue Interest with the Zipf-Mandelbrot distribution [33], and the malicious users issue Interest with uniform distribution. In Zipf-Mandelbrot distribution, the content items with k-th rank in the whole content popularity ranking list are requested with probability $\{q_k\}_{k=1,2\ldots K}$, where $q_k = c/(k+q)^s$, $c = \left\{\sum_{k=1}^{K} 1/(k+q)^s\right\}^{-1}$, K is the size of the popularity list, and q and s are parameters.

4.1. Performance Metrics. The performance of detection mechanism is evaluated by the confusion matrix, as shown in Figure 11, where TP represents the number of abnormal traffic, which is classified as abnormal, TN represents the number of normal traffic, which is classified as normal, FP represents the

Table 3: Simulation parameters.

Parameters	Value
Legitimate request distribution	Zipf-Mandelbrot
Malicious request distribution	Uniform
Number of content types	1000
Malicious request rate	100
Legitimate request rate	100
Lifetime of PIT entries (second)	1
Attack time (second)	400
Simulation time	800

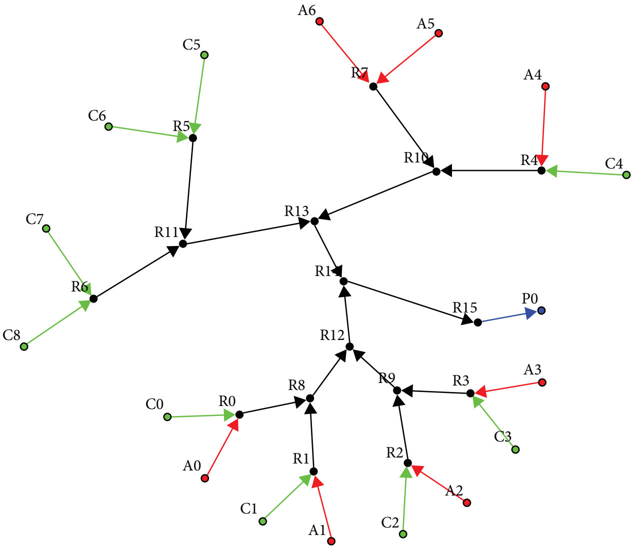


Figure 10: Tree topology.

number of normal traffic, which is classified as abnormal, and FN represents the number of abnormal traffic, which is classified as normal. This paper compares the detection mechanism with SVM and LSTM from the following metrics:

(i) Interest satisfaction ratio: it is defined as the ratio between the number of Data packets received and the number of Interest packets sent.

(ii) PIT size: it is defined as the number of entries in the PIT.

(iii) Accuracy: it is defined as the overall performance of the model and is calculated as follows:

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}. \tag{10}$$

(iv) Recall: it is defined as the proportion of attack samples that are correctly identified as attacks, and it is calculated as follows:

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{11}$$

4.2. Hyperparameter Tuning. The detection model's architectures are built using Pytorch in Python on a machine with 32 GB RAM. This paper trains detection model for 50 epochs with Adam optimizer [34] at a learning rate of 0.001.

4.3. Loss Function. The binary cross entropy is a loss function that is used in binary classification problems. The objective of the detection mechanism is to label time window as normal or abnormal; therefore, this paper uses binary cross entropy loss function for training the LSTM and LSTM with attention mechanism, which is computed as follows:

$$L = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log\left(p\left(y_i\right)\right) + \left(1 - y_i\right) \cdot \log\left(1 - p\left(y_i\right)\right). \tag{12}$$

where y_i is the binary label and N is the total number of samples in training set.

4.4. Impact of the IFA. Attack intensity (λ) is defined as the ratio of malicious user's sending rate to the legitimate user's sending rate. In this section, this paper evaluates the impact of the IFA and considers two types of routers: the router only connected to legitimate user and the router connected to legitimate user and malicious user.

In Figure 10, this paper evaluates PIT size of the routers $R11$, $R10$, and $R8$ under IFA and evaluates the Interest satisfaction ratio of normal users under the IFA.

Figure 12 shows the PIT size under IFA with different attack intensities. When there is no attack, the routers have a constant PIT size. When IFA is launched at the 400th second, the PIT size begins to increase and the greater the attack intensity, the greater the PIT size. The impact on PIT size is also different for routers in different locations; the router R11 is least affected by the attack because it is not connected to a malicious user; the router R10 is greatly affected by the attack because it has the most connections with malicious users.

Figure 13 shows the Interest satisfaction ratio of normal user under IFA with different attack intensities. The Interest satisfaction ratio is stable without IFA, and the Interest packet sent by the user can receive the corresponding Data packet. At the 400th second, the IFA is launched, the Interest packets sent by users can hardly receive the corresponding Data packets, and the Interest satisfaction ratio decreases instantaneously. Moreover, with the increase of attack intensity, more malicious Interest packets are sent and the impact on Interest satisfaction ratio of normal users is greater.

4.5. Performance of Detection Mechanism. In this section, this paper compares our detection mechanism with SVM and LSTM from detection accuracy and recall. Then, this paper evaluates the defend mechanism from Interest satisfaction ratio and PIT size with expired-PIT-based defend mechanism [35].

Firstly, the learning rate and batch size used in this paper are introduced. This paper selects learning rate and batch size by comparing the detection accuracy. The learning rate is 0.001, 0.005, and 0.01, respectively. The batch size is 512, 256, and 128, respectively. The simulation results of different learning rates and batch sizes on the detection accuracy are shown in Figures 14–16, respectively.

	normal	abnormal
normal	TN	FP
abnormal	FN	TP

Figure 11: Confusion matrix.

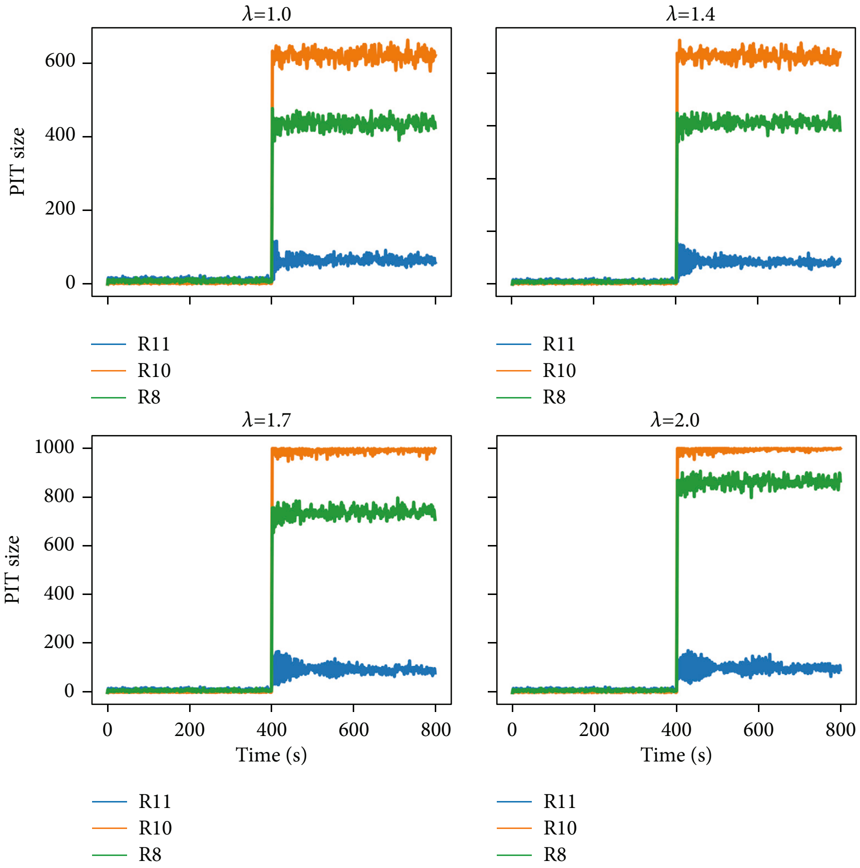


Figure 12: PIT size under IFA.

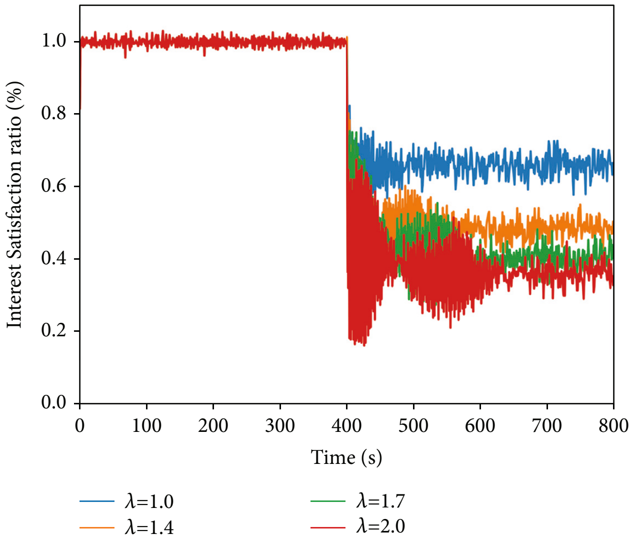


Figure 13: Interest satisfaction ratio under IFA.

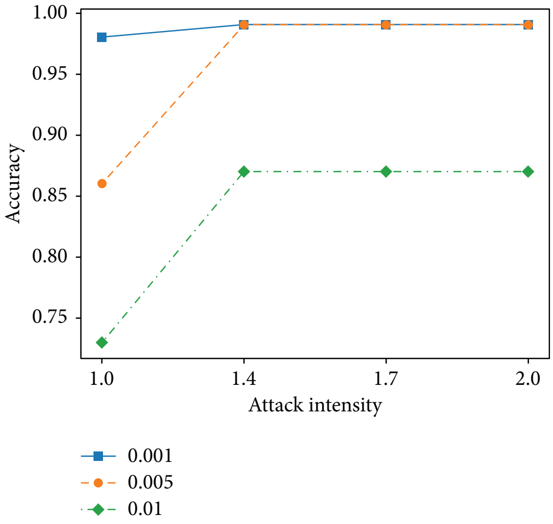


Figure 14: Batch size 512.

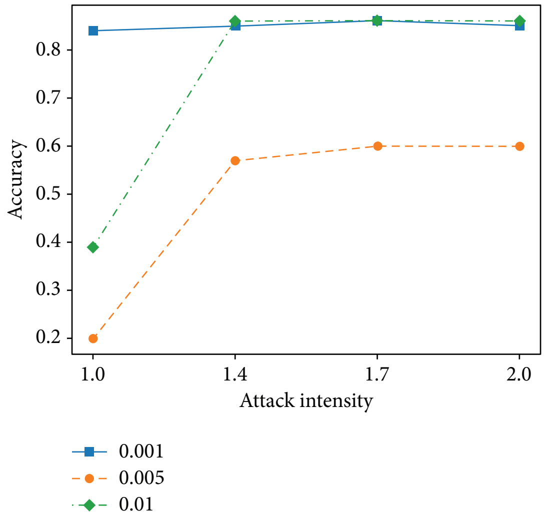


Figure 15: Batch size 256.

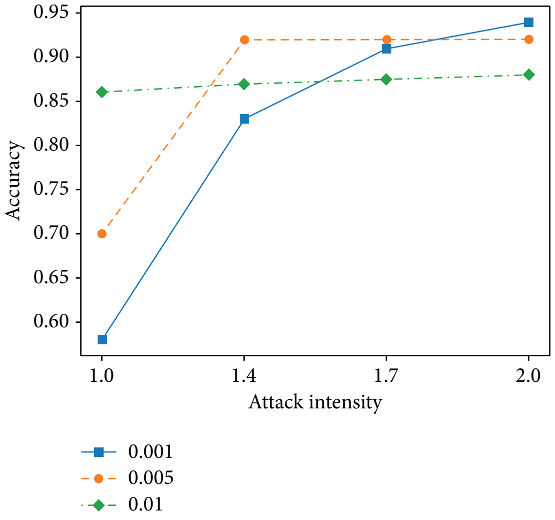


Figure 16: Batch size 128.

Figure 14 shows the detection accuracy under different attack intensities with different learning rates when the batch size is 512. When the learning rate is 0.001, the accuracy is the highest.

Figure 15 shows the detection accuracy under different attack intensities with different learning rates when the batch size is 256. When the learning rate is 0.001, the accuracy is the highest.

Figure 16 shows the detection accuracy under different attack intensities with different learning rates when the batch size is 128.

Finally, this paper sets the batch size 512 and the learning rate is 0.001. As shown in Figures 17 and 18, with the increase in the number of epochs, the accuracy increases and the loss decreases. When the epochs are equal to 50, the model tends to be stable.

Next, this paper compares the accuracy and recall of the detection mechanism with SVM and LSTM, and the results are shown in Figures 19 and 20.

Figure 19 shows the detection accuracy of the proposed detection mechanism under different attack intensities. Compared with LSTM and SVM, the detection mechanism proposed in this paper has the highest accuracy.

Figure 20 shows the recall of the proposed detection mechanism under different attack intensities. Compared with LSTM and SVM, the detection mechanism proposed in this paper has the highest recall.

4.6. Performance of Mitigation Mechanism. This section evaluates our mitigation mechanism on the Interest satisfaction ratio and PIT size.

Figure 21 shows the Interest satisfaction ratio with the proposed defend mechanism and expired-PIT-based defend mechanism under attack. When the malicious users launch IFA at the 400th second, the Interest satisfaction ratio drops rapidly. Under high attack intensity, the proposed detection mechanism quickly detects the attack and limits the sending of malicious packets and the Interest satisfaction ratio returns to the normal level. This paper also tests the impact of the detection mechanism on the burst traffic of normal users, and the proposed detection mechanism will not misjudge the burst traffic of normal users.

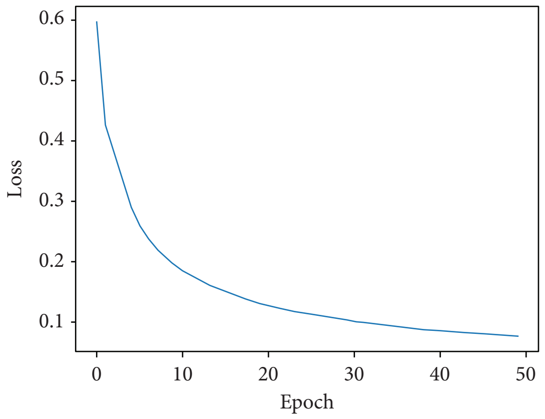


Figure 17: Loss with epoch.

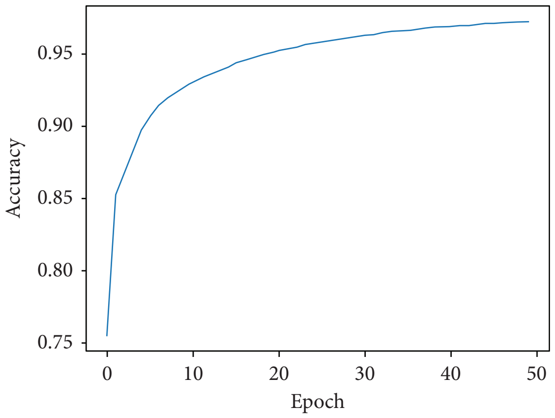


Figure 18: Accuracy with epoch.

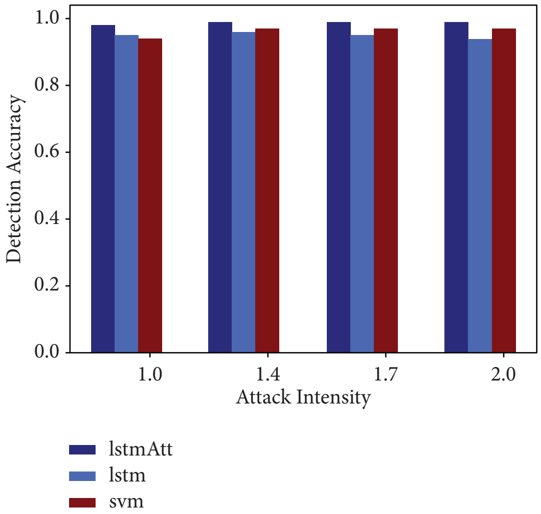


Figure 19: Detection accuracy.

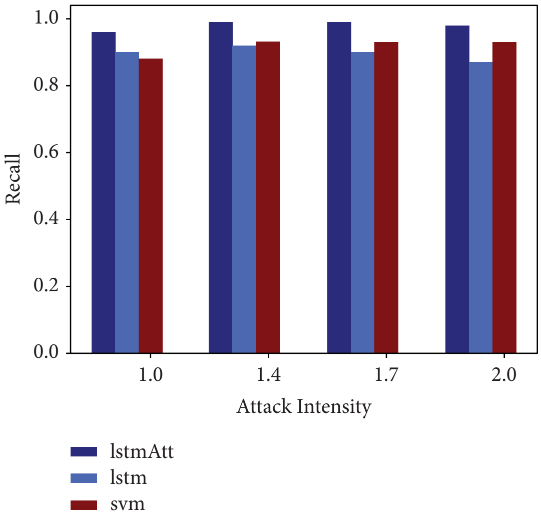


Figure 20: Recall.

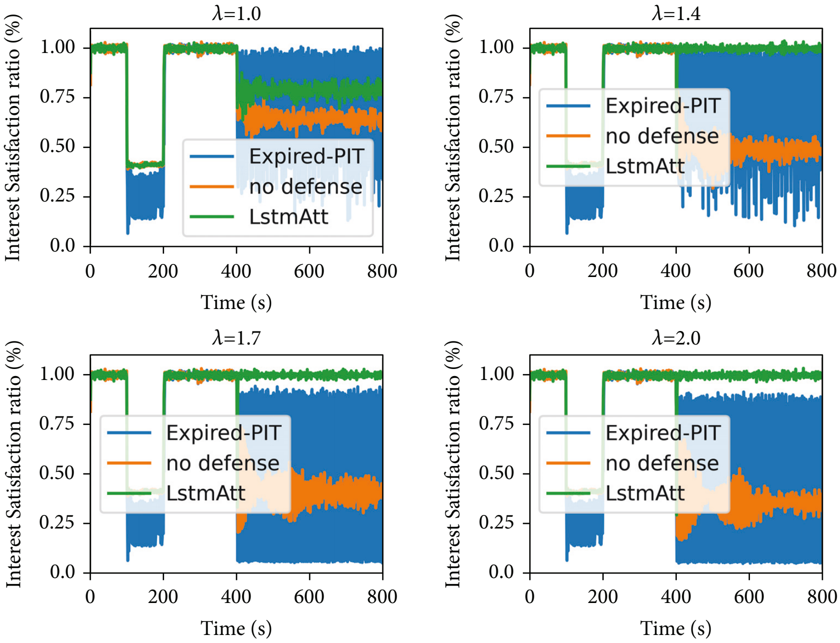


Figure 21: Interest satisfaction ratio with different defend mechanisms.

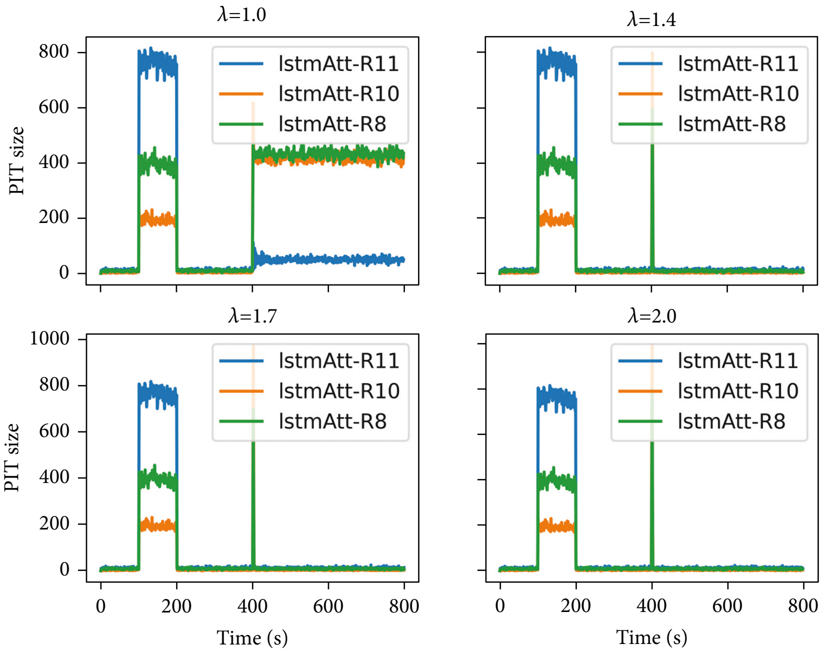


Figure 22: PIT size with the proposed defend mechanism.

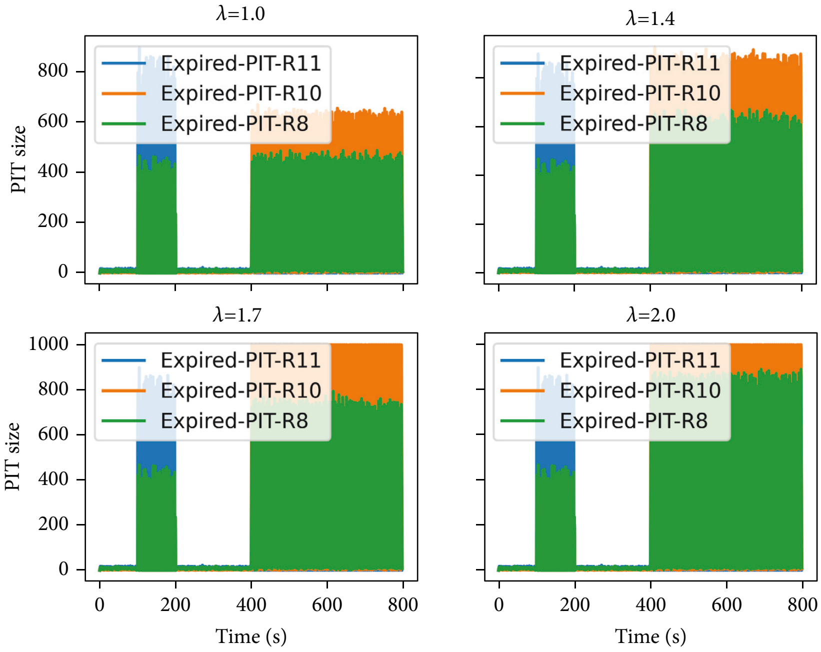


Figure 23: PIT size with expired-PIT-based defend mechanism.

Figures 22 and 23 show the PIT size with the proposed defend mechanism and expired-PIT-based defend mechanism under attack. When the attacker starts the attack at the 400th second, the PIT size rises rapidly. Under high attack intensity, the detection mechanism quickly detects the attack of different attack intensities and limits the sending of malicious packets and the PIT size returns to the normal level.

5. Conclusions

This paper proposes a defend mechanism for Interest flooding attack in NDN. The defend consists of three parts: detection, response, and mitigation. The LSTM with attention mechanism is used to detect IFA; once IFA is detected, the Hellinger distance is used to identify malicious Interest packet prefix. Finally, the malicious prefix is sent to the downstream routers to cooperate to limit the attack. The experimental results show that the LSTM with attention mechanism shows better performance than the LSTM and SVM. In future work, this paper will consider multiple attacks in NDN, such as collusive attack, low-rate IFA, and large-scale topology.

Data Availability

The data used to support the findings of this study have not been made available because the data also form part of an ongoing study.

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 (61862046); Huhhot Science & Technology Plan (2021-KJXM-TZJW-04); and the Science and Technology Program of Inner Mongolia Autonomous Region (2020GG0188).

References

[1] G. Xylomenos, C. N. Ververidis, V. A. Siris et al., “A survey of information-centric networking research,” *IEEE communications surveys & tutorials*, vol. 16, no. 2, pp. 1024–1049, 2014.

[2] B. Ahlgren, M. D’ambrosio, and C. Dannewitz, “Second netinf architecture description,” *4WARD EU FP7 Project*, Deliverable D-6.2 v2. 0, 2010.

[3] M. Ain, D. Trossen, and P. Nikander, “D2. 3–architecture definition, component descriptions, and requirements,” *Deliverable*, PSIRP 7th FP EU-funded project, vol. 11, 2009.

[4] T. Koponen, M. Chawla, B.-G. Chun et al., “A data-oriented (and beyond) network architecture,” *s* in *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communication*, vol. 37, no. 4, pp. 181–192, Kyoto, Japan, August 2007.

[5] V. Jacobson, M. Mosko, D. Smetters, and J. Garcia-Luna-Aceves, “Content-centric networking,” *Palo Alto Research Center*, White Paper, pp. 2–4, 2007.

[6] L. Zhang, A. Afanasyev, J. Burke et al., “Named data networking,” *ACM SIGCOMM - Computer Communication Review*, vol. 44, no. 3, pp. 66–73, 2014.

[7] H. Zhang, Y. Li, Z. Zhang, A. Afanasyev, and L. Zhang, “NDN host model,” *ACM SIGCOMM - Computer Communication Review*, vol. 48, no. 3, pp. 35–41, 2018.

[8] P. Gasti, G. Tsudik, E. Uzun, and L. Zhang, “DoS and DDoS in named data networking,” in *Proceedings of the 2013 22nd International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–7, Nassau, Bahamas, August 2013.

[9] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[10] A. Basu, A. Mandal, and L. Pardo, “Hypothesis testing for two discrete populations based on the Hellinger distance,” *Statistics & Probability Letters*, vol. 80, no. 3-4, pp. 206–214, 2010.

[11] T. Zhi, Y. Liu, J. Wang, and H. Zhang, “Resist interest flooding attacks via entropy-SVM and jensen-shannon divergence in information-centric networking,” *IEEE Systems Journal*, vol. 14, no. 2, pp. 1776–1787, 2020.

[12] G. Xing, J. Chen, R. Hou et al., “Isolation forest-based mechanism to defend against interest flooding attacks in named data networking,” *IEEE Communications Magazine*, vol. 59, no. 3, pp. 98–103, 2021.

[13] J. Zhou, J. Luo, L. Deng, and J. Wang, “Defense mechanism of interest flooding attack based on deep reinforcement learning,” in *Proceedings of the 2020 3rd International Conference on Hot Information-Centric Networking (HotICN)*, pp. 65–70, Hefei, China, December 2020.

[14] N. Kumar, A. K. Singh, and S. Srivastava, “Feature selection for interest flooding attack in named data networking,” *International Journal of Computers and Applications*, vol. 43, no. 6, pp. 537–546, 2021.

[15] N. Kumar, A. K. Singh, and S. Srivastava, “Evaluating machine learning algorithms for detection of interest flooding attack in named data networking,” in *Proceedings of the 10th International Conference on Security of Information and Networks*, pp. 299–302, Jaipur, India, October 2017.

[16] Z. Wu, R. Zhang, and M. Yue, “A method for joint detection of attacks in named data networking,” *Journal of Computer Research and Development*, vol. 58, no. 3, pp. 569–582, 2021.

[17] T. Zhi, H. Luo, and Y. Liu, “A Gini impurity-based interest flooding attack defence mechanism in NDN,” *IEEE Communications Letters*, vol. 22, no. 3, pp. 538–541, 2018.

[18] R. Hou, M. Han, J. Chen et al., “Theil-based countermeasure against interest flooding attacks for named data networks,” *IEEE Network*, vol. 33, no. 3, pp. 116–121, 2019.

[19] Z. Wu, W. Feng, M. Yue, X. Xu, and L. Liu, “Mitigation measures of collusive interest flooding attacks in named data networking,” *Computers & Security*, vol. 97, Article ID 101971, 2020.

[20] Y. Nakatsuka, J. L. Wijekoon, and H. Nishi, “FROG: a packet hop count based DDoS countermeasure in NDN,” in *Proceedings of the 2018 IEEE Symposium on Computers and Communications (ISCC)*, pp. 00492–00497, Natal, Brazil, June 2018.

[21] J. Dong, K. Wang, W. Quan, and H. Yin, “InterestFence: simple but efficient way to counter interest flooding attack,” *Computers & Security*, vol. 88, Article ID 101628, 2020.

[22] Y. Xin, Y. Li, W. Wang, W. Li, and X. Chen, “Detection of collusive interest flooding attacks in named data networking using wavelet analysis,” in *Proceedings of the 2017 IEEE Military Communications Conference (MILCOM)*, pp. 557–562, Baltimore, MD, USA, October 2017.

[23] A. Benarfa, M. Hassan, E. Losiouk, and A. M. B. M. Compagno, “ChoKIFA+: an early detection and mitigation approach against interest flooding attacks in NDN,” *International Journal of Information Security*, vol. 20, no. 3, pp. 269–285, 2021.

[24] D. Qu, G. Lv, S. Qu, and H. Y. Z. Shen, “An effective and lightweight countermeasure scheme to multiple network attacks in NDN,” *IEEE/ACM Transactions on Networking*, vol. 30, no. 2, pp. 515–528, 2022.

[25] T. Nguyen, H.-L. Mai, R. Cogranne et al., “Reliable detection of interest flooding attack in real deployment of named data networking,” *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 9, pp. 2470–2485, 2019.

[26] A. Benarfa, M. Hassan, A. Compagno, E. Losiouk, M. B. Yagoubi, and M. Conti, “ChoKIFA: a new detection and mitigation approach against interest flooding attacks in

NDN," , Springer, Bologna, Italy, 2019pp. 53–65, Lecture Notes in Computer Science, vol. 11618.

[27] K. Cho, B. V. Merrienboer, C. Gulcehre et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, https://arxiv.org/abs/1406.1078.

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

[29] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, https://arxiv.org/abs/1409.0473.

[30] G. Zhang, V. Davoodnia, A. Sepas-Moghaddam, Y. Zhang, and A. Etemad, "Classification of hand movements from EEG using a deep attention-based LSTM network," *IEEE Sensors Journal*, vol. 20, no. 6, pp. 3113–3122, 2020.

[31] Y. Ding, Y. Zhu, J. Feng, P. Zhang, and Z. Cheng, "Interpretable spatio-temporal attention LSTM model for flood forecasting," *Neurocomputing*, vol. 403, pp. 348–359, 2020.

[32] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM: NDN simulator for NS-3," 2012, https://named-data.net/wp-content/uploads/TRndnsim.pdf.

[33] Z. K. Silagadze, "Citations and the Zipf-Mandelbrot's law," 1999, https://arxiv.org/abs/physics/9901035.

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

[35] V. G. Vassilakis, B. A. Alohali, I. D. Moscholios, and M. D. Logothetis, "Mitigating distributed denial-of-service attacks in named data networking," in *Proceedings of the 11th Advanced International Conference on Telecommunications (AICT)*, pp. 18–23, Brussels, Belgium, June 2015.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 1028251, 14 pages
https://doi.org/10.1155/2022/1028251

Research Article

Detection of Packet Dropping Attack Based on Evidence Fusion in IoT Networks

Weichen Ding,[1] **Wenbin Zhai**,[1] **Liang Liu**,[1] **Ying Gu**,[2] **and Hang Gao**[1]

[1] *College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China*
[2] *School of Engineering and Applied Sciences, Columbia University, New York, NY, USA*

Correspondence should be addressed to Liang Liu; liangliu@nuaa.edu.cn

Received 7 April 2022; Revised 6 June 2022; Accepted 16 June 2022; Published 9 July 2022

Academic Editor: Kai Wang

Internet of Things (IoT) is widely used in environmental monitoring, smart healthcare, and other fields. Due to its distributed nature, IoT is vulnerable to various internal attacks. One of these attacks is the packet-dropping attack, which is very harmful. The existing packet-dropping attack detection algorithms are unsuitable for emerging resource-constrained IoT networks. For example, ML-based algorithms always inject numerous packets to obtain the training dataset. However, it is heavyweight for energy-limited nodes to forward these extra packets. In this paper, we propose a lightweight evidence fusion-based detection algorithm (EFDA), which leverages the packet forwarding evidence to identify malicious nodes. Firstly, EFDA finds the sequence numbers of dropped packets and their corresponding source nodes. Then, it traces the routing path of each dropped packet and collects evidence for detection. The evidence stored by nodes around the path record the node's forwarding behaviors. Finally, the collected evidence is fused to evaluate the trust of nodes. Based on nodes' trust, the K-means clustering is used to distinguish between malicious nodes and benign nodes. We conduct simulation experiments to compare EFDA with ML-based algorithms. The experimental results demonstrate that EFDA can detect the packet-dropping attack without injecting packets and achieve a higher detection accuracy.

1. Introduction

In the last decade, the Internet of things (IoT) has become a popular infrastructure to support many applications, such as intelligent transportation [1] and smart home [2]. IoT is a system consisting of interrelated computing devices, which collect and process the data acquired from the environments. These devices (such as sensors) cooperate with each other through the IoT protocol, including ZigBee [3], Wi-Fi [4], and Bluetooth.

With the rapid development and application of IoT, it is prone to varied attacks, among which the packet-dropping attack is very hard to detect and prevent. In the packet-dropping attack, malicious attackers can invade and control legitimate devices to discard some essential packets halfway, causing the base station loses the important information. For example, malicious nodes drop the vital packets in the healthcare wireless sensor network (WSN) that contains the alarm information for the patient's health parameters such as blood pressure and heart rate [5]. If the alarm information is not transmitted to the doctors but dropped halfway, the patients will be at risk. It is vital to detect malicious nodes.

1.1. Motivation. In recent years, many traditional packet-dropping attack detection algorithms have been proposed, but they are not suitable for the emerging resource-constrained IoT networks. For instance, traditional machine learning (ML)-based detection algorithms [6–9] identify malicious nodes by training detection models. The performance of the detection models depends on the size of the training dataset. To get a large size of the training dataset, numerous labeled packets need to be injected into the IoT networks. However, it is heavyweight for energy-limited nodes to forward numerous injected packets. It is crucial to propose a lightweight algorithm for resource-constrained

IoT networks. To overcome this problem, we propose a lightweight evidence fusion-based detection algorithm (EFDA), which uses the packet forwarding evidence (PFEs) to identify malicious nodes. The PFE is generated during the packet forwarding process. When a node in the network forwards a packet, it locally stores a packet forwarding record. Due to the broadcast characteristic of wireless communication in IoT networks, each neighbor of the node can sniff the packet and generate a PFE.

As shown in Figure 1, EFDA contains three phases.

(1) Getting the dropped packet set: the base station needs to find the dropped packets and their corresponding source nodes. For this purpose, the base station divides received packets into groups according to their source nodes. Then, the base station sorts the received packets in each group based on their sequence numbers. The dropped packets can be found because their sequence numbers are not in the groups. Figure 1 shows an example that the base station divides the received packets into two groups according to two source nodes: N_1 and N_2. After sorting the packets in each group, it finds that the dropped packet is N_1.Packet$_2$, which is the identifier of the packet whose corresponding source node is N_1.

(2) Collecting PFEs: for each dropped packet, the base station traces its routing path and finds the suspicious nodes. In Figure 1, the base station sends a request to N_1 to ask it for the next forwarding node of N_1.Packet$_2$; N_1 searches its forwarding records and finds that the next forwarding node of N_1.Packet$_2$ is N_2, on behalf of the base station, N_1 asks N_2 for the next forwarding node of the packet; N_2 searches its forwarding records and finds that next forwarding node is N_4; N_2 continues to ask N_4 for the next forwarding node of the packet; N_4 reports the next forwarding node of the packet is N_6. But after N_4 asks N_6, N_6 reports that it has never received the packet. At this moment, the base station finds a logic conflict between N_4 and N_6, and then, it identifies N_4 and N_6 as suspicious nodes. To resolve the logic conflict and find the liar, the base station collects PFEs stored by the neighbors of N_4, namely N_2, N_3, and N_5.

(3) Fusing PFEs: the base station fuses the collected PFEs. Because N_2, N_3, and N_5 provide PFEs to prove that N_4 has forwarded N_1.Packet$_2$ to N_6, the base station discovers that N_6 lies to hide its dropping packet behavior.

As mentioned above, EFDA does not need to inject extra packets to obtain the training dataset to train the detection model, and it utilizes the existing PFEs in the network to perform logical reasoning and identify malicious nodes.

In summary, the contributions of this paper are as follows.

We propose a lightweight evidence fusion based packet dropping attack detection algorithm (EFDA) for the resource-constrained IoT networks. EFDA uses the packet forwarding evidences to detect malicious nodes, which achieves a high detection accuracy with a low cost. We conduct simulation experiments to systematically evaluate our detection algorithm. The experimental results show that EFDA provides better detection accuracy than ML-based algorithms.

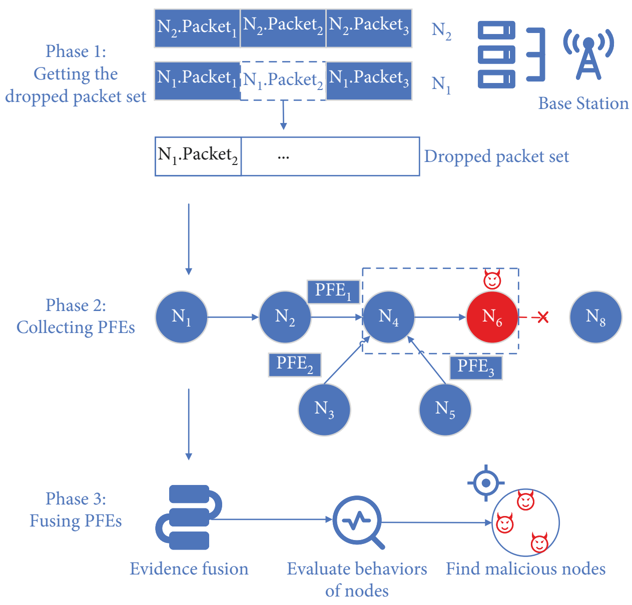


Figure 1: The process of EFDA.

1.2. Organization. The remainder of this paper is organized as follows. Section 2 introduces the related work of the packet dropping attack detection in IoT. Section 3 formalizes the packet dropping attack. Section 4 details our detection algorithm, EFDA. Section 5 shows the results of the simulation experiments. Section 6 concludes this paper.

2. Related Work

To resist the packet dropping attack, a wide variety of algorithms are proposed in recent researches, which can be divided into five categories: monitor-based algorithms, acknowledgment-based algorithms, camouflage-based algorithms, ML-based algorithms, and other algorithms.

2.1. Monitor-Based Algorithm. The core of monitor-based algorithms is to place monitoring nodes among communication nodes and classify them into "normal" and "abnormal" by the collected traffic data [10]. Watchdog [11] is a basic technology for the packet dropping attack detection, where a monitoring node sniffs the traffic of the next hop to detect the attacks of malicious nodes. Li et al. [12] applied the watchdog to monitor the behavior of nodes rather than traffic data, and they detected malicious nodes by comparing the interval of sending and receiving packets with the threshold. In the monitor-based detection (CMD) [13], each node monitors the packet loss rates of its preferred parent node and its one-hop neighbor nodes. By comparing the packet loss rate of its preferred parent and one-hop neighbor nodes, the monitoring node can find the abnormal behaviors of its preferred parent node.

2.2. Acknowledgment-Based Algorithms. The acknowledgment-based algorithms depend on the acknowledgment (ACK) packet to detect malicious nodes [14]. Each node is responsible for monitoring the forwarding behaviors of its next node and reporting it to the base station by sending ACK packets. In adaptive acknowledgment-based approach (AAA) [15], each node monitors its one-hop and two-hop downstream nodes. After forwarding a data packet, the node overhears the forwarding behavior of its one-hop downstream node and waits to receive an ACK packet from its two-hop downstream node. Once receiving no ACK packet, the node identifies its one-hop downstream node as a malicious node. In single checkpoint-based detection (SCAD) [16], the source node randomly selects an intermediate node on the routing path as the checkpoint node for each packet. After receiving the packet, both the sink node and checkpoint node need to reply an ACK packet to the source node. If receiving no ACK packet, other intermediate nodes will send an alarm packet to the source node to suspect their downstream nodes, which are identified as malicious nodes.

2.3. Camouflage-Based Algorithms. In the energy harvesting motivated networks (EHNets), some nodes called energy harvesting node need to periodically harvest energy from an immediate environment. In camouflage-based active detection (CAM) [17], each node actively disguises it as an energy harvesting node and pretends not to overhear its adjacent nodes. But actually, each node monitors any forwarding behaviors of its adjacent nodes. Once finding abnormal behaviors, they identify that adjacent node as a malicious node. In the EYES [18], each node not only actively disguises itself as an energy harvesting node to overhear the forwarding behaviors of its adjacent nodes but also validates any previous uncertain forwarding behavior to detect malicious nodes.

2.4. ML-Based Algorithms. Machine learning (ML) is a common and efficient technology, which has been widely used in malicious node detection. Akbani et al. [19] combined the ML with the reputation systems (RS), which automates the process of designing the RS model. Liu et al. [20] proposed a trust system, which calculated the trust of each node by the trust of each routing path. Based on the trusts of nodes, they were divided into malicious or benign group. Liu et al. [21] improved this scheme, and they used the method of linear regression to calculate the trust of nodes, which was more accurate than [20]. Also, they took into account the possibility that nodes launched the multiple-mix-attack. Yang et al. [22] considered a more fine-grained attack named selective-edge packet attack, and they argued that malicious nodes may be more intelligent to launch an attack selectively. Also, they selected the best scheme after sifting through various types of regression algorithms and clustering algorithms.

2.5. Other Algorithms. In [23], due to most of the detection algorithms are for the centralized networks, blockchain-based multimobile code-driven trust mechanism (BMCTM) is proposed to detect malicious nodes in decentralized networks. It combines the blockchain technology and trust system, which detects nodes as malicious nodes according to their low trusts. A secure routing framework is proposed in [24], which leverages a new type of packet called dummy packet to detect malicious nodes. The dummy packet scheme is used to find the critical routes and detect malicious nodes in the critical routes. In [25], considering malicious nodes may lie to attract and drop packets during route establishment phase, and a robust hybrid method is proposed to strengthen the route security.

Most of the above algorithms are heavyweight for the emerging resource-constrained IoT networks. For monitor-based algorithms, acknowledgment-based algorithms, and camouflage-based algorithms, the energy-limited nodes need to monitor the forwarding behaviors of their neighbor nodes and to converge collected data all the times. For ML-based algorithms, the energy-limited nodes need to assist them to obtain the training dataset by forwarding numerous injected packets. They are heavyweight for energy-limited nodes. Therefore, in this paper, we propose a lightweight evidence fusion-based detection algorithm (EFDA) to achieve a high detection accuracy.

3. Network and Attack Model

In this section, the network model is introduced, and the packet-dropping attack is formalized. Table 1 exhibits a list of notations for later reference.

3.1. Network Model. In this paper, the IoT network is a multihop wireless network consisting of sensor nodes, which communicate with each other through the routing protocol for low-power and lossy networks (RPL). The sensor nodes collect data and encapsulate them into packets. The packets are forwarded by relay nodes to the base station. A typical IoT network is shown in Figure 2.

A node is represented as N_i ($i \in [1, M]$), and the base station is represented as S. Each node has at least one routing path to the base station S. A routing path is represented as Path_j ($j \in [1, K]$), which is expressed as

$$\text{Path}_j = [N_1 \longrightarrow N_2 \longrightarrow \cdots \longrightarrow N_i \longrightarrow S], \tag{1}$$

where it represents a packet which is sent from N_1, forwarded through $N_2, \ldots, N_i$ in a sequence, and finally received by the base station S.

Then, the network is expressed as

$$\begin{aligned} &\text{Network} = (N, S, P),\\ &\quad N = \{N_1, N_2, \ldots, N_i, \ldots, N_M\},\\ &\quad P = \{\text{Path}_1, \text{Path}_2, \ldots, \text{Path}_j, \ldots, \text{Path}_K\}, \end{aligned} \tag{2}$$

where N is the set of nodes in the network, and P is the set of routing paths in the network, Network is the network

Table 1: Notations.

Symbol	Meaning
N	The set of nodes in a network
N_i	A node in N
S	The base station in a network
P	The set of routing paths in a network
Path_j	A routing path in P
$N_i.\text{Packet}_k$	The identifier of a packet whose source node is N_i
P_d	The attack probability of a node

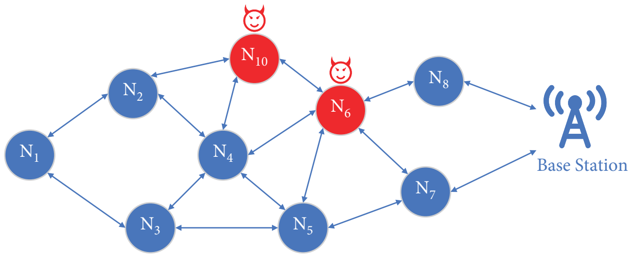


Figure 2: A typical IoT network.

consisting of the sensor nodes, the base station, and the routing paths.

3.2. Packet Dropping Attack Model. If there are no malicious nodes, a packet will arrive at the base station. However, the packet may be discarded halfway if there are malicious nodes.

As shown in Figure 3, the malicious node N_6 drops the packet $N_1.\text{Packet}_2$. In this paper, the malicious nodes may launch the packet-dropping attack with a certain probability P_d ($P_d \in (0,1]$). We use $N_i.P_d$ to represent the probability that N_i launches a packet-dropping attack. Considering the harmfulness of the packet-dropping attack and the constrained resources of the IoT network, the malicious nodes should be detected with a high accuracy and a low overhead.

3.3. PFE Model. The packet forwarding evidences (PFEs) are generated during the packet transmission. Due to the broadcast characteristic of wireless communication, when a node N_f forwards a packet $N_i.\text{Packet}_j$, all neighbors of N_f can sniff the packet. The receiving node N_r will receive the packet, and other neighbors of N_f generate PFEs to record the forwarding behavior of N_f. The generated PFE can be represented as

$$\text{PFE} = \left(N_f, N_i.\text{Packet}_j, N_r\right), \tag{3}$$

where it represents the neighbors of N_f witness that N_f has forwarded the packet $N_1.\text{Packet}_j$ to N_r.

As shown in Figure 4, N_4 wants to forward the packet $N_1.\text{Packet}_2$ to N_6. Due to the broadcast characteristic of wireless communication, all neighbors of N_4 can sniff the packet. N_6 receives the packet, and the other neighbors (N_2, N_3, N_5, N_{10}) of N_4 generate a PFE, namely $(N_4, N_1.\text{Packet}_2, N_6)$.

During packet transmission, each node generates numerous PFEs according to the forwarding packet behaviors of its neighbors. We design a table named PFE Table (PFET) for each node to store the PFEs. PFET is shown in Table 2.where there are four fields: *Packet-ID*, *Forwarding Node*, *Receiving Node*, and *Capacity*. *Packet-ID* means the identifier of the forwarded packet; *Forwarding Node* means the node that forwards the packet; *Receiving Node* means the node that receives the packet; *Capacity* means the number of PFEs that a node can store. We assume that a node's total capacity is C, and it is divided equally to its neighbors. According to Table 2, we can know that N_{10} has generated three PFEs about N_4, which, respectively, represent N_4 has forwarded $N_1.\text{Packet}_2$ to N_6, $N_1.\text{Packet}_3$ to N_5, and $N_2.\text{Packet}_1$ to N_3.

To avoid PFE being faked or tampered, we apply the signcryption in [26] to transfer the PFE. The signcryption generalized-CLSC (gCLSC) is secure and lightweight, which can be used in the resource-constrained IoT network. Before sending a PFE to the base station, the sending node encrypts and signs the PFE with gCLSC. After receiving the encrypted and signed PFE, the base station verifies the sending node's signature and decrypts the PFE with gCLSC.

4. Algorithm

In this section, we introduce our evidence fusion-based detection algorithm (EFDA), which is divided into three phases. (1) Getting the dropped packet set: the base station finds the dropped packet set and the source node of each dropped packet. (2) Collecting PFEs: for each dropped packet, the base station traces its routing path and finds the suspicious nodes. PFEs stored by neighbors of suspicious nodes are collected. (3) Fusing PFEs: the base station fuses the collected PFEs to detect malicious nodes.

4.1. Getting the Dropped Packet Set. The source nodes collect data from the environment, encapsulate them into packets, and then upload the packets to the base station. After receiving the packets, the base station divides the received packets into different groups G_i ($i \in [1, M]$) according to their source nodes N_i. For each group, the base station sorts the packets according to their sequence numbers. After grouping and sorting the received packets, the base station can find the dropped packets and their corresponding source nodes.

As Figure 5 shows, the base station divides the received packets into M groups and sorts the packets for each group. For the first group of the source node N_1, the base station receives the packets with sequence number $N_1.\text{Packet}_1$, $N_1.\text{Packet}_2$, and $N_1.\text{Packet}_4$ except $N_1.\text{Packet}_2$. So it finds that $N_1.\text{Packet}_2$ is dropped. After checking all groups, the base station can obtain the dropped packet set.

4.2. Collecting PFEs. After finding all the dropped packets, the base station traces the routing path of each dropped packet. In the process of tracing, the base station investigates the nodes on the routing path hop by hop. In the final hop, it

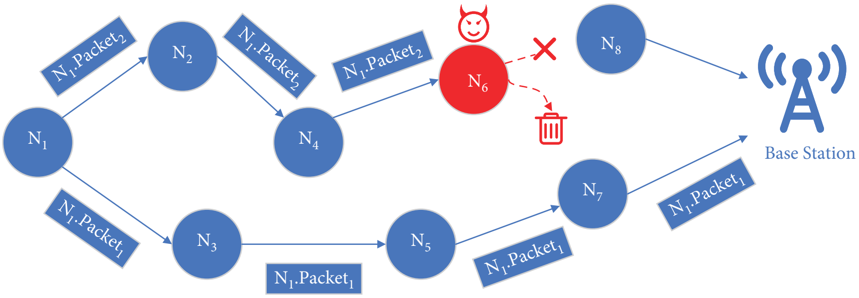


Figure 3: A network with malicious nodes.

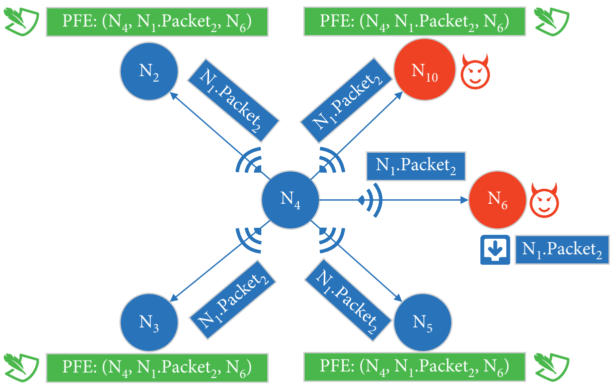

PFE: (N_4, N_1.Packet$_2$, N_6)
PFE: (N_4, N_1.Packet$_2$, N_6)
N_2
N_1.Packet$_2$
N_1.Packet$_2$
N_{10}
N_1.Packet$_2$
N_4
N_6
N_1.Packet$_2$
N_1.Packet$_2$
N_1.Packet$_2$
N_3
N_5
PFE: (N_4, N_1.Packet$_2$, N_6)
PFE: (N_4, N_1.Packet$_2$, N_6)

Figure 4: An example of the PFEs generated.

Table 2: PFET of N_{10}.

Packet-ID	Forwarding node	Receiving node	Capacity
N_1.Packet$_2$	N_4	N_6	$C/2$
N_1.Packet$_3$		N_5	
N_2.Packet$_1$		N_3	
N_4.Packet$_1$	N_6	N_5	$C/2$
N_4.Packet$_2$		N_7	
N_5.Packet$_1$		N_7	

can find two suspicious nodes that may drop the packet. To judge the suspicious nodes, PFEs about them are collected at the base station. We propose an evidence collection protocol (ECP) to trace the routing path and collecting PFEs.

In order to assist ECP to trace the routing path of each dropped packet, each node in the network needs to generate records about its packet forwarding behaviors. Each node maintains a packet forwarding record table (PFRT) to store the records, which is shown in Table 3. It contains three fields: *Last Node*, *Packet-ID*, and *Next Node*. *Last Node* means the last node that forwards the packet, *Packet-ID* means the identifier of the forwarded packet, and *Next Node* means the next node where the packet is forwarded. After a node receives a packet and forwards the packet to another node, it will update its PFRT to record the forwarding behavior.

As shown in Figure 6, during the transmission of the packet N_1.Packet$_2$, N_4 receives the packet from N_2 and forwards it to N_6. To record this forwarding behavior, N_4 inserts a record $(N_2, N_1.\text{Packet}_2, N_6)$ into its PFRT. Besides, malicious nodes may not update their PFRTs because they drop packets instead of forwarding them.

Based on the packet forwarding records stored by nodes, ECP can trace the routing path of each dropped packet. For a dropped packet N_i.Packet$_j$, the process of tracing the packet can be described as follows.

The base station finds the dropped packet N_i.Packet$_j$ and its corresponding source node N_i. It constructs a TM message (shown in Table 4) *tm* {"Packet-ID": "N_i.Packet$_j$"} and sends it to N_i. The message *tm* is used to ask N_i for the next forwarding node of N_i.Packet$_j$. After that, the base station initializes the tracing progress as $[N_i]$. Once receiving *tm*, node N_i searches its PFRT for the packet forwarding record about the packet. It finds the next forwarding node is N_s. It constructs a RM message (shown in Table 5) *rm* {"Successor": "N_s"} and sends it to the base station. The message *rm* is used to report the tracing progress to the base station. Besides, N_i forwards *tm* to N_s to ask it to continue to trace the routing path of the packet. When the base station receives *rm*, it updates the tracing progress as $[N_i \longrightarrow N_s]$. After receiving *tm*, node N_s repeats the operations like N_i to continue to trace the routing path. After several steps of tracing, the tracing progress is updated to $[N_i \longrightarrow N_s \longrightarrow \cdots \longrightarrow N_k \longrightarrow N_m]$, and a malicious node N_m receives *tm*.

As shown in Figure 7, the base station finds the dropped packet N_1.Packet$_2$ and its corresponding source node N_1. Then, it sends a TM message *tm* to N_1 and initializes the tracing progress as $[N_1]$. Once receiving *tm*, node N_1 searches its PFRT and finds the next forwarding node is N_2. It sends a RM message rm_1 to the base station and forwards *tm* to N_2. When the base station receives rm_1, it updates the tracing process as $[N_1 \longrightarrow N_2]$. Once receiving *tm*, node N_2 continues to trace the routing path of the packet. After several steps of tracing, the tracing progress is updated to $[N_1 \longrightarrow N_2 \longrightarrow N_4 \longrightarrow N_6]$, and N_6 receives *tm* from N_4. We assume that node N_6 is a malicious node.

When a malicious node N_m receives *tm*, there are three possible cases as follows.

Case 1. The malicious node does not respond to the base station.

The continued tracing process is described as follows.

After receiving *tm*, the malicious node N_m does not respond to the base station. Once receiving no response from N_m, the base station identifies N_m as a malicious node.

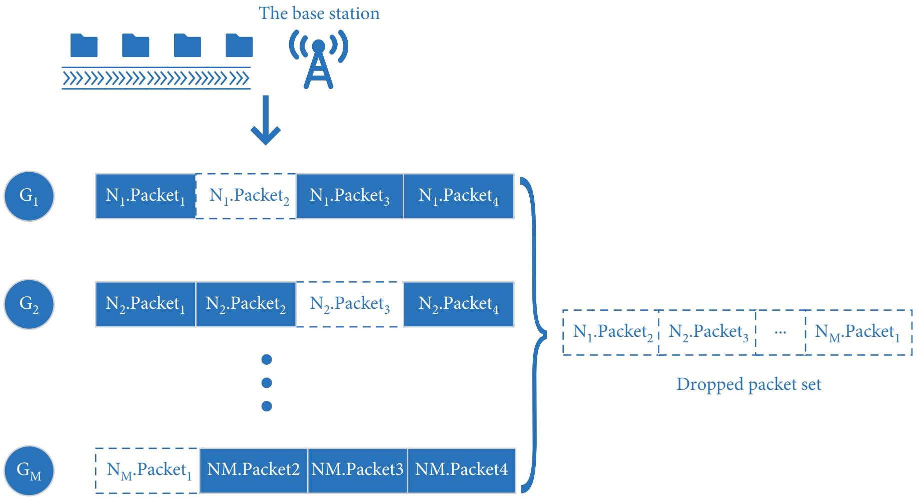
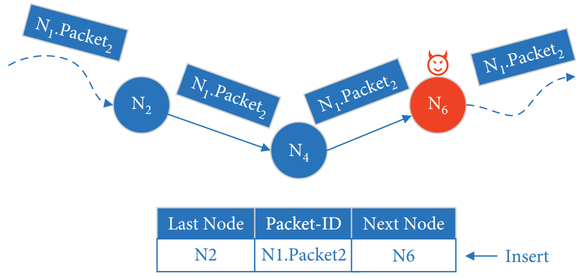


Figure 5: The process of getting the dropped packets set.

Table 3: Packet forwarding record table of N_4.

Last node	Packet-ID	Next node
N_2	N_1.Packet$_2$	N_6



Last Node	Packet-ID	Next Node
N2	N1.Packet2	N6

Figure 6: An example of updating the PFRT.

Table 4: Tracing message (TM).

Field	Description
Packet-ID	The identifier of a dropped packet

As shown in Case 1 of Figure 7, N_6 has received *tm* from N_4, but it does not respond to the base station. The base station identifies N_6 as a malicious node.

Case 2. The malicious node responds that it has never received the packet forwarded by its predecessor.

The continued tracing process is described as follows:

After receiving *tm*, the malicious node N_m denies that it has received N_i.Packet$_j$ from N_k. So, it constructs an IM message (shown in Table 6) *im* {"Packet-ID": "N_i.Packet$_j$," "Impeaching-Node": "N_m," "Impeached-Node": "N_k"} and sends it to the base station. The message *im* is used to report the base station that N_m has never received N_i.Packet$_j$ from N_k. When the base station receives *im*, it finds a logic conflict between N_k and N_m. It identifies N_k and N_m as suspicious nodes.

Table 5: Reporting message (RM).

Field	Description
Successor	The next forwarding node of the dropped packet

As shown in Case 2 of Figure 7, N_6 denies that it has received N_1.Packet$_2$ from N_4. Then, it sends an IM message im_1 to the base station. After receiving im_1, the base station identifies N_4 and N_6 as suspicious nodes.

Case 3. The malicious node responds that it has forwarded the packet to a neighbor, but actually not.

The continued tracing process is described as follows:

After receiving *tm*, the malicious node N_m lies that it has forwarded N_i.Packet$_j$ to N_n. So, it sends *rm* {"Successor": "N_n"} to the base station. Besides, it forwards *tm* to N_n. When the base station receives *rm*, it updates the tracing progress as $[N_i \longrightarrow N_s \longrightarrow \cdots \longrightarrow N_k \longrightarrow N_m \longrightarrow N_n]$. Once receiving *tm*, node N_n searches its PFRT but finds no packet forwarding record about N_i.Packet$_j$. So, it sends *im* {"Packet-ID": "N_i.Packet$_j$," "Impeaching-Node": "N_n," "Impeached-Node": "N_m"} to the base station to deny that it has received N_i.Packet$_j$ from N_m. When the base station receives *im*, it finds a logic conflict between N_m and N_n. It identifies N_m and N_n as suspicious nodes.

As shown in Case 3 of Figure 7, N_6 lies that the next forwarding node is N_8. It sends rm_4 to the base station and forwards *tm* to N_8. When the base station receives rm_4, it updates the tracing progress as $[N_1 \longrightarrow N_2 \longrightarrow N_4 \longrightarrow N_6 \longrightarrow N_8]$. Once receiving *tm*, node N_8 sends im_2 to the base station to deny that it has received N_1.Packet$_2$ from N_6. When the base station receives im_2, it identifies N_6 and N_8 as suspicious nodes.

After the tracing process of a dropped packet like N_i.Packet$_j$, the base station can get two suspicious nodes like N_k and N_m (N_m and N_n). To find the liar in them, the base station needs to collect PFEs about them as follows.

Without loss of generality, we suppose the suspicious nodes are N_k and N_m.

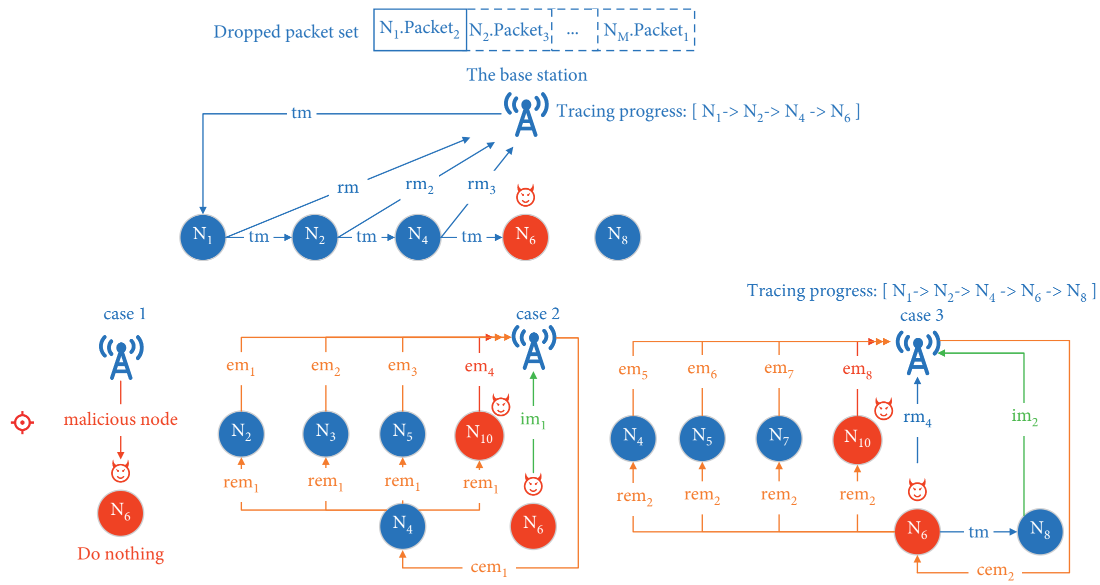


Figure 7: An example of ECP.

Table 6: Impeaching message (IM).

Field	Description
Packet-ID	The identifier of a dropped packet
Impeaching-node	The node impeaches that impeached node has not forwarded the packet to it
Impeached-node	The node is impeached

Table 7: Collecting evidences message (CEM).

Field	Description
Expected-PFE	The PFE is expected by the base station

Table 8: Requesting evidence message (REM).

Field	Description
Requested-PFE	The PFE is requested by the sending node

The base station constructs a CEM message (shown in Table 7) *cem* {“Expected-PFE”: “$(N_k, N_i.\text{Packet}_j, N_m)$”} and sends it to the precursor N_k of the suspicious nodes. The message *cem* is used to collect PFEs. Once receiving *cem*, N_k constructs a REM message (shown in Table 8) *rem* {“Expected-PFE”: “$(N_k, N_i.\text{Packet}_j, N_m)$”} and sends it to its neighbors. The message *rem* is used to request the neighbors to send the relevant PFEs to the base station. When each neighbor of N_k receives *rem*, it searches its PFET for the PFE. Once getting the matched PFE, each neighbor constructs an EM message (shown in Table 9) *em* {“PFE”: “$(N_k, N_i.\text{Packet}_j, N_m)$”} and sends it to the base station. But, the accomplice does not submit correct PFE by sending *em* {"PFE": “(No Evidence)”} or submits a faked PFE to the base station. After receiving all EM messages, the base station extracts the PFEs in them.

As shown in Case 2 of Figure 7, the base station finds the suspicious nodes N_4 and N_6. It sends a CEM message cem_1 to the precursor N_4. Once N_4 receives cem_1, it sends a REM message rem_1 to its neighbors. When the neighbors (N_2, N_3, N_5, and N_{10}) receive rem_1, benign neighbor N_2 (N_3, N_5) searches its PFETs and sends the matched PFE by an EM message em_1 (em_2, em_3) {“PFE”: $N_4, N_1.\text{Packet}_2$, N_6“$(N_4, N_1.\text{Packet}_2, N_6)$”} to the base station, but the accomplice N_{10} sends em_4 {“PFE”: “(No Evidence)”} to the base station. After receiving all *EM* messages, the base station extracts all PFEs.

After tracing the routing path of a dropped packet by ECP, the base station can get two suspicious nodes and their relevant PFEs.

4.3. Fusing PFEs

After the base station finds the suspicious nodes and collects their relevant PFEs by ECP, we propose an evidence fusion algorithm (EFA) to fuse these PFEs and detect malicious nodes.

The evidence fusion is actually a voting process. For the suspicious nodes N_k and N_m about $N_i.\text{Packet}_j$, N_k’s neighbors send either PFE (N_k, $N_i.\text{Packet}_j$, N_m) or PFE (No Evidence) to the base station. PFE (N_k, $N_i.\text{Packet}_j$, N_m) means a neighbor witnesses N_k has forwarded $N_i.\text{Packet}_j$ to N_m, and it votes for N_k. PFE (No Evidence) means a neighbor regards N_k as the liar, and it votes for N_m. The number of votes for a node is represented as v, and v_k (v_m) is the number of votes for N_k (N_m). Malicious neighbors may submit a faked PFE to vote for its accomplices. To mitigate the effects of the collusion among malicious nodes, we use nodes’ weights to multiply nodes’ votes. The weight of N_i is represented as ω_i, and it is the ratio of N_i’s trust to the initial value, namely $\omega_i = t_i/T$. The trust of N_i $(i \in [1, M])$ is represented as t_i, and the initial value of t_i is T. The base station maintains a trust and weight table (TWT) to record the trusts and weights of all nodes.

Table 9: Evidence message (EM).

Field	Description
PFE	The stored PFE is encrypted and signed by the sending node, and it sends to the base station

As the votes are weighted, the number of votes for a node is the sum of the weights of the neighbors that have voted for it. After fusing PFEs, the base station identifies the node with fewer votes as the liar and punishes it by decreasing its trust.

For the dropped packet N_1.Packet$_2$ in Case 2 of Figure 7, the base station gets four PFEs $\{N_2\colon (N_4, N_1.\text{Packet}_2, N_6)$, $N_3\colon (N_4, N_1.\text{Packet}_2, N_6)$, $N_5\colon (N_4, N_1.\text{Packet}_2, N_6)$, $N_{10}\colon (\text{No Evidence})\,\}$. N_2, N_3, and N_5 witness that N_4 has forwarded N_1.Packet$_2$ to N_6, and they regard N_4 as a benign node and vote for it. But N_{10} denies that N_4 has forwarded the packet to N_6, and it regards N_6 as a benign node and votes for it. Because ω_2, ω_3, ω_5, and ω_{10} are initialized as 1, $v_4 = \omega_2 + \omega_3 + \omega_5 = 3$ and $v_6 = \omega_{10} = 1$. By comparing v_4 with v_6, the base station identifies N_6 as the liar and decreases its trust.

For a dropped packet, ECP traces its routing path to find two suspicious nodes and the relevant PFEs, and EFA fuses these PFEs to discover the liar and decreases its trust. After ECP traces all dropped packets and EFA punishes all liars over, the final trusts of nodes can be obtained. Based on the final nodes' trusts, the K-means clustering is used to cluster nodes into two groups: malicious group (MG) and benign group (BG).

As shown in Algorithm 1, EFDA contains three steps.

(1) Getting the dropped packet set: the base station divides the received packets into different groups G_i according to their source nodes (line 3–5) and sorts the packets according to their sequence numbers for each group (line 6-7). After grouping and sorting the received packets, the base station can find the dropped packet set, namely DPS (line 8–12).

(2) Collecting PFEs: for each dropped packet, the base station S sends a TM message to the source node N_i to start a tracing process (line 14–15). The current node N_{cur} finds the successor N_s and continues the tracing process until the base station finds two suspicious nodes N_k, N_m (line 17–23). The base station collets PFEs about N_k, N_m to judge them (line 24–28).

(3) Fusing PFEs: for two suspicious nodes, the base station fuses their relevant PFEs to update their votes (line 29–35). The node with fewer votes is punished by decreasing its trust (line 36–41). Based on the final nodes' trusts, the K-means clustering is used to cluster nodes to BG and MG (line 43).

4.4. Algorithm Analysis

4.4.1. Algorithm Complexity Analysis. According to the pseudocode in Algorithm 1, the proposed approach contains three steps: (1) getting the dropped packet set: in order to get the dropped packet set, EFDA needs to traverse the received packet set (RPS). The complexity of the first step is $O_1 = \text{size}(\text{RPS})$, where size(RPS) means to find the size of the set RPS. (2) Collecting PFEs: in order to collect PFEs, EFDA needs to traverse the dropped packet set (DPS). For each dropped packet, EFDA needs to trace the routing path of the dropped packet. The complexity of the second step is $O_2 = \text{size}(\text{DPS}) \times \text{size}(\text{Path})$, where Path means the traced routing path. (3) Fusing PFEs: in order to fuse PFEs, EFDA needs to traverse the PFEs for each dropped packet. The complexity of the third step is $O_3 = \text{size}(\text{DPS}) \times \text{size}(\text{PFEs})$. Therefore, the complexity of EFDA is represented as $O = O_1 + O_2 + O_3 = \text{size}(\text{RPS}) + \text{size}(\text{DPS}) \times \text{size}(\text{Path}) + \text{size}(\text{PFEs})$.

4.4.2. Algorithm Overheads Analysis

(1) Energy Overheads of EFDA. EFDA detects malicious nodes by tracing the routing paths of dropped packets. In addition, it can get the detection results in a limited number of dropped packets. We assume that the limited number of dropped packets is L. For each tracing process, considering the worst case, all nodes on the routing path need to send a TM message and a RM message. Only some specified nodes need to send an IM message, a REM message, or an EM message. Therefore, each node sends no more than 3 extra messages for one tracing process. Because EFDA needs to trace L dropped packets, each node sends no more than $3 \times L$ extra messages for one time detection. Moreover, since the sizes of the above messages are small, the energy overheads of sending them are small.

(2) Storage Overheads of EFDA. In EFDA, each node needs to store the packet forwarding records and PFEs, and we estimate the storage overheads of EFDA to prove its feasibility. The storage overheads of a node are affected by the sizes of its PFRT and PFET. A general IoT device forwards 1200 messages/minute according to the study in [27]. Assuming that EFDA is executed every hour to detect malicious nodes. During this period, there are 72000 messages forwarded and 72000 packet forwarding records stored by a node. For a packet forwarding record, it contains three fields, and its storage overheads are 3 Bytes. The storage overheads of PFRT are $72000 \times 3 = 216000\,\text{Bytes} \approx 210\,\text{kB}$. For PFET, its capacity C (shown in Table 2) is approximate to the number of forwarded packets, namely 72000. For a PFE, it contains three fields, and its storage overheads are 3 Bytes. The storage overheads of PFET are $72000 \times 3 = 216000\,\text{Bytes} \approx 210\,\text{kB}$. So, the storage overheads of a node are $210 + 210 = 420\,\text{kB}$, which is far less than a general IoT device's storage 2 GB [28].

4.4.3. Distinction between EFDA and ML-Based Algorithms. In this section, we analyze the distinction between EFDA and ML-based algorithms. For each injected packet, ML-based algorithms use it to calculate the trust of its routing path by mathematical reasoning. Then, they use the

```
Input: RPS (Received Packet Set)
Output: BG (Benign Group), MG (Malicious Group)
Initialize BG = $\emptyset$, MG = $\emptyset$;
Step1 Getting the dropped packet set:
Initialize all $G_i = \emptyset$, DPS = $\emptyset$ ;
foreach $N_i$.Packet$_j \in$ RPS do
    $G_i = G_i \cup N_i$.Packet$_j$;
end
for $i = 1; i \leq M; i$ ++ do
    $G_i$ sorts inner $N_i$.Packet$_j$ in ascending order of $j$;
    for $j = 1; j \leq j_{\max}; j$ ++ do
        if $N_i$.Packet$_j \notin G_i$ then
            $DP\ S = DP\ S \cup N_i$.Packet$_j$;
        end
    end
end
Step2 Collecting PFEs:
foreach $N_i$.Packet$_j \in DP\ S$ do
    $S \longrightarrow {}^{\mathrm{TM}} N_i$, Process $= \{N_i\}$;
    $N_{cur} = N_i$;
    while Find no suspicious nodes do
        $N_{cur}$ finds next forwarding node is $N_s$;
        $N_{cur} \longrightarrow {}^{\mathrm{RM}} S$, $N_{cur} \longrightarrow^{T} M\ N_s$;
        $N_{cur} = N_s$;
        Process $=$ Process $\cup N_s$;
    end
    $S$ finds two suspicious nodes $(N_k, N_m)$;
    $S \longrightarrow {}^{\mathrm{CEM}} N_k$;
    $N_k \longrightarrow {}^{\mathrm{REM}} N_k's$ neighbors;
    foreach neighbor $\in N_k's$ neighbors do
        neighbor $\longrightarrow {}^{\mathrm{PFE}} S$;
    end
Step3 Fusing PFEs:
    foreach PFE $\in$ PFEs do
        if $PFE == \{N_k, N_i$.Packet$_j, N_m\}$ then
            $V_k$ ++;
        else
            $V_m$ ++;
        end
    end
    if $V_k > V_m$ then
        Decrease $N_m$'s trust;
    end
    if $V_k < V_m$ then
        Decrease $N_k$'s trust;
    end
end
Based on nodes' trusts, K-means clusters nodes to BG and MG;
return (BG, MG);
```

Algorithm 1: EFDA algorithm.

routing path's trust to estimate the nodes' trusts on the routing path. However, in order to estimate the nodes' trusts more accurately, numerous packets need to be injected to get more routing path's trusts, which are used as the input of the ML-based algorithms.

On the contrary, EFDA detects malicious nodes without injecting packets. It can trace the routing path of each dropped packet and find the suspicious nodes. The PFEs around suspicious nodes are collected to the base station, and EFDA fuses them to find the malicious nodes. A potential constraint for EFDA is how to resist collusive attacks. Suppose that a benign node is surrounded by many malicious nodes, they submit faked PFEs that cause the base station to misidentify the benign node as a malicious node. A possible extension is to use the causal inference algorithm to solve the problem.

Table 10: Experimental evaluation.

		Detection result		
		Negative	Positive	Total
Actual result	Negative	True positive (TP)	False negative (FN)	P (actual negative)
	Positive	False positive (FP)	True negative (TN)	N (actual positive)
	Total	P' (detect negative)	N' (detect positive)	$P + N$

Table 11: Variables and description.

Variables	Description
The number of uploaded packets	The number of packets that are uploaded to the base station will influence the detection accuracy
The number of nodes	It means the number of nodes deployed in the network, which can affect the scale of the network and the detection accuracy
The percentage of malicious nodes	It means that how many nodes are malicious in the network, which can affect the detection result
The probability of attack	Malicious nodes launch the packet-dropping attack with a probability, and less probability means that the node is more difficult to be detected. It can influence the detection accuracy
The diversity of network	It essentially indicates the ratio of available routing paths that could be chosen by source nodes to upload packets. The diversity of network reflects the routing paths' complexity, and it influences the detection accuracy

5. Performance Evaluation

In this section, we evaluate the performance of our proposed EFDA and compare it with two typical ML-based algorithms, namely HD [20] and PDE [21].

Both HD and PDE need to inject numerous labeled packets into the network and collect them at the base station. Each labeled packet has a routing path, and each routing path has abundant labeled packets. For each routing path, not all labeled packets on the routing path can be collected by the base station due to the malicious nodes. HD and PDE define the trust of the routing path as a ratio, which is the number of collected labeled packets to the total number of labeled packets on the routing path. According to whether a node is on the routing path, the relationship between the trust of nodes and the trust of the routing path can be formalized as a mathematical equation. The mathematical equation can be solved by machine learning algorithms, and the trust of nodes can be obtained. Based on the trust of nodes, the clustering algorithm classifies them into benign group and malicious group.

We evaluate accuracy and error rate to compare detection performance. As shown in Table 10, the accuracy is defined as $P_a = (\mathrm{TP} + \mathrm{TN})/(P + N)$, and the error rate is defined as $F_a = (\mathrm{FP})/(\mathrm{FP} + \mathrm{TN})$.

5.1. Experimental Environment

5.1.1. Environmental Settings. In our environment, all nodes are evenly distributed in a rectangle area of $100 \times 100\,\mathrm{m}^2$, and each node's communication range is 10 m. Our IoT network is generated randomly, and there is at least one routing path from each source node to the base station.

To avoid bias, we run our simulation for each experiment in 10 rounds with 10 different networks generated randomly. The average value of 10 rounds' result is calculated as the final experimental result of each experiment. In particular, we use the simulator in [21] and add our EFDA to it. Both EFDA and the ML-based algorithms are deployed at the base station.

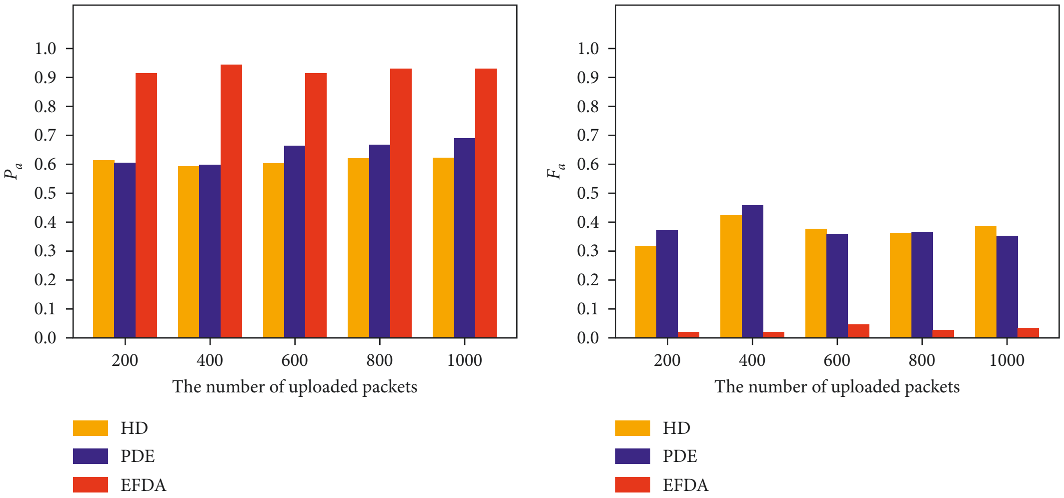
5.1.2. Environmental Variables. In the following experiments, we investigate the impact of the variables (shown in Table 11) on the detection performance. Unless otherwise specified, all experimental variables will remain the default, which is set as follows.

The number of uploaded packets is 500. The number of nodes is 15. The probability of an attack is 0.3. The percentage of malicious nodes is 0.3. The diversity of the network is 1.

5.2. HD vs PDE vs EFDA.

In this section, we explore the performance comparison among HD, PDE, and EFDA through experiments.

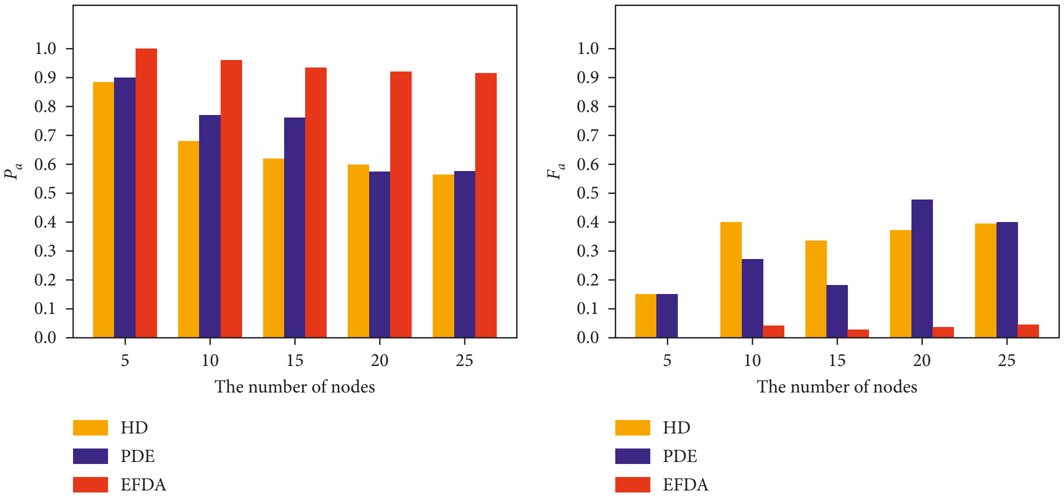
5.2.1. Impact of the Number of Uploaded Packets. The results in Figure 8 show that EFDA performs better than HD and PDE. When the number of uploaded packets is small, HD and PDE get a low P_a. As the number of uploaded packets increases, HD and PDE can get a higher P_a. EFDA gets a stale P_a in all cases. This is because as the number of uploaded packets increases, HD and PDE can calculate more routing path's trust to estimate nodes' trusts. Once more collected information is used to estimate nodes' trusts, HD and PDE can get more exact nodes' trusts and get more accurate detection results. EFDA can trace the routing path to find the suspicious nodes and detect malicious nodes in a smaller detection range. EFDA hardly needs abundant



(a)

(b)

Figure 8: The impact of the number of uploaded packets. (a) Impact of the number of uploaded packets on P_a. (b) Impact of the number of uploaded packets on F_a.



(a)

(b)

Figure 9: The impact of the number of nodes. (a) Impact of the number of nodes on P_a. (b) Impact of the number of nodes on F_a.

routing path's information to estimate nodes' trusts, so it can get a stable detection results.

5.2.2. Impact of the Number of Nodes. The results in Figure 9 show that when the number of nodes is small, all algorithms get a high P_a and a low F_a; but when the number of nodes increases, the accuracy P_a of all algorithms decreases, and the error rate F_a of them increases. EFDA still performs better than HD and PDE in all cases. This is because when the number of nodes is 5, the network topology is simple, and malicious nodes are more easily to be detected; when the number of nodes increases and the network topology becomes more complex, the malicious nodes are more likely to hide their abnormal behaviors, and it is difficult to identify all malicious nodes. However, no matter how complex the network topology becomes, EFDA still reaches higher accuracy than HD and PDE.

5.2.3. Impact of the Percentage of Malicious Nodes. The results in Figure 10 show that EFDA gets the better results than the other two detection algorithms; but with the percentage of malicious nodes increases, the accuracy P_a of EFDA is getting lower and the error rate F_a of EFDA is getting higher, while the trends of HD and PD remain stable. This is because when the percentage of malicious nodes increases, the number of malicious nodes in the network will also increase that leads to more malicious nodes cooperate to resist EFDA. Assuming that most of the neighbors around a benign node are malicious, the malicious neighbors vote for its accomplice, which causes the benign node to be

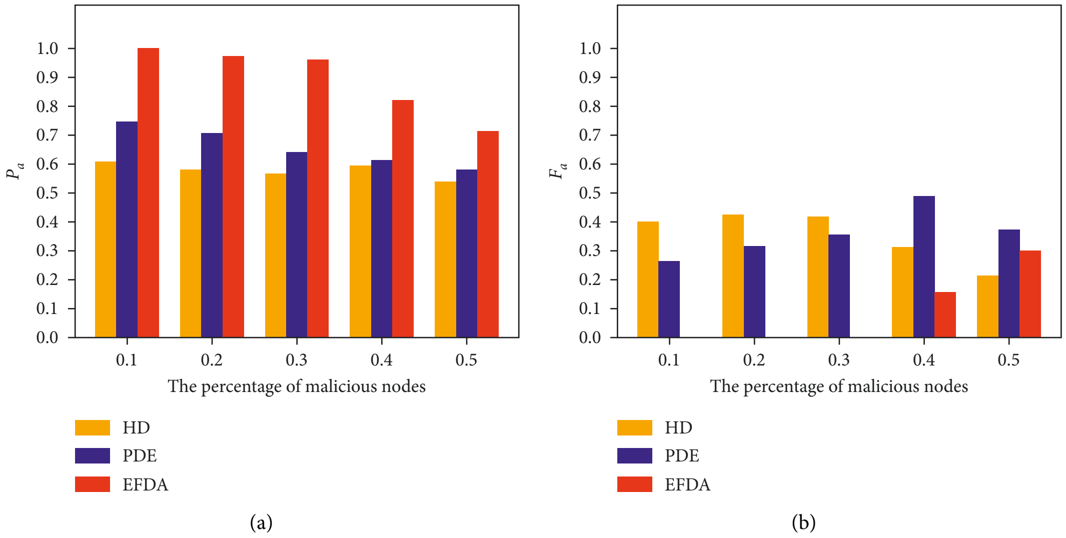


Figure 10: The impact of the percentage of malicious nodes. (a) Impact of the percentage of malicious nodes on P_a. (b) Impact of the percentage of malicious nodes on F_a.

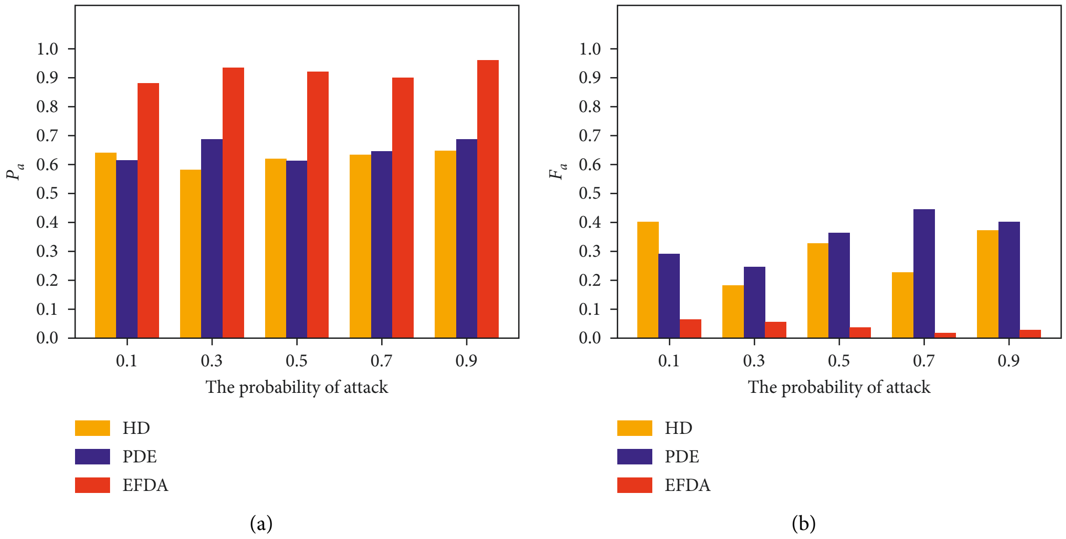


Figure 11: The impact of the probability of attack. (a) Impact of the probability of attack on P_a. (b) Impact of the probability of attack on F_a.

misidentified as the liar, and its trust is decreased by EFDA. Because EFDA misidentified the benign node as a malicious node, it gets a lower P_a and a higher F_a. Although we have optimizations for the collusion among malicious nodes, it is difficult to resist the collusion attacks from many malicious nodes.

5.2.4. Impact of the Probability of Attack. The results in Figure 11 show that EFDA performs better than HD and PDE. When the probability of attack is small, EFDA gets a small P_a and a large F_a. However, when the probability of attack increases, the accuracy P_a of EFDA begins to increase, and the error rate F_a of EFDA begins to decrease. The trends of HD and PDE are similar, but their accuracy P_a is lower than that of EFDA, and their error rate F_a is higher than that of EFDA. This is because when the probability of attack is small, malicious nodes intend to hide their attack behaviors that make EFDA more difficult to detect them. However, when the probability of attack becomes larger, malicious nodes are more likely to launch a packet dropping attack that makes EFDA find more dropped packets. EFDA traces more routing paths of the dropped packets and finds more suspicious nodes, and it gets more accurate detection results.

5.2.5. Impact of the Diversity of Network. The results in Figure 12 show that when the diversity of network is low, both HD and PDE get a low P_a and a high F_a. With the diversity of network increases, their accuracy P_a becomes higher, and their error rate F_a becomes lower. However, EFDA gets stable accuracy P_a and error rate F_a in all cases, and they are better than those of HD and PDE. This is because when the diversity of network is low, there are few

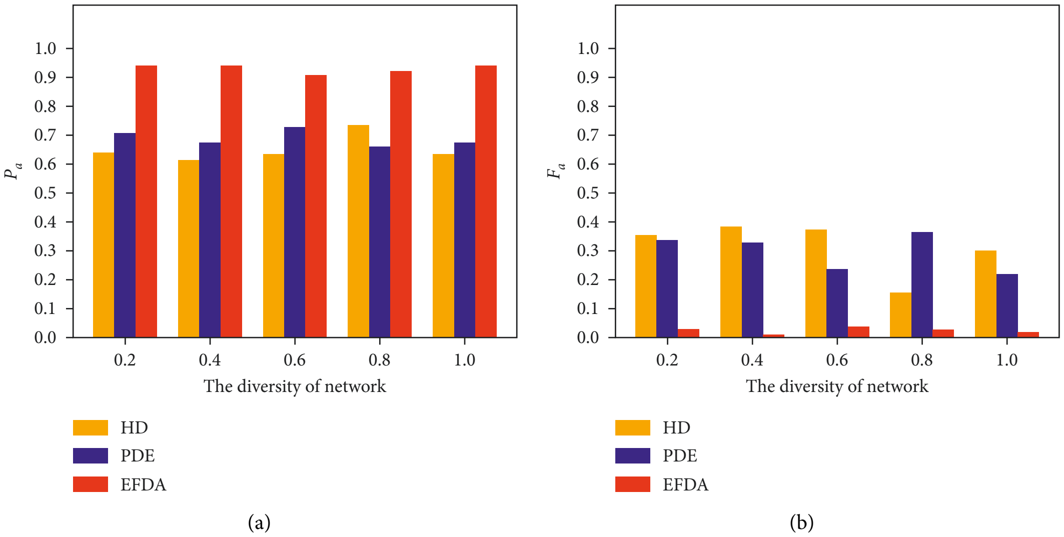


Figure 12: The impact of the diversity of network. (a) Impact of the diversity of network on P_a. (b) Impact of the diversity of network on F_a.

routing paths for source nodes to upload packets to the base station. It means HD and PDE obtain few routing paths' information to estimate the nodes' trusts, and that causes HD and PDE to get the inaccurate nodes' trusts. Therefore, they get negative detection results. As the diversity of network becomes larger, HD and PDE obtain more routing paths' information to estimate the nodes' trusts, and they get positive detection results. However, EFDA does not need more different routing paths' information to estimate the nodes' trusts. It can trace the path of dropped packet and accurately find the suspicious nodes on the path, and it only decreases the liar's trust. So, EFDA detects malicious nodes more efficiently than HD and PDE.

5.3. Discussion and Limitations. In the experiments, we explore the performance comparison between HD, PDE, and EFDA on five variables, which are the number of uploaded packets, the number of nodes, the percentage of malicious nodes, the probability of attack, and the diversity of the network. Overall, it is observed that EFDA can achieve better detection performance compared with HD and PDE. EFDA can improve the detection rate by around 20% to 30%.

Although EFDA performs better than HD and PDE, there are some limitations that can be addressed in our future work. When the percentage of malicious nodes exceeds 50%, the detection performance of EFDA declines significantly, which indicates that EFDA is difficult to resist the collusion of numerous malicious nodes. In our future work, we plan to investigate how to resist the collusion of numerous malicious nodes.

6. Conclusion

Due to the distributed nature of the IoT networks, they are vulnerable to the packet-dropping attack. There are abundant detection algorithms to detect the packet dropping attack; however, most of them are heavyweight for the resource-constrained IoT network. In this paper, we propose a lightweight evidence fusion-based detection algorithm, namely EFDA. It uses packet forwarding evidence to detect malicious nodes. In EFDA, the received packets are grouped and sorted to find the dropped packets. For each dropped packet, the base station traces its routing path, finds the suspicious nodes, and collects evidence. The collected evidences are fused to find the liar, and EFDA punishes the liar by decreasing its trust. Based on nodes' trusts, the K-means clustering is used to cluster nodes and detect malicious nodes.

Our experimental results demonstrate that EFDA has better detection performance than two typical ML-based algorithms: HD and PDE. EFDA detects malicious nodes without injecting packets, and it can improve the detection accuracy by around 20% to 30%.

Data Availability

Some or all data, models, or codes generated or used during the study are available from the corresponding author by request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under no. U20B2050 Public Service Platform for Basic Software and Hardware Supply Chain Guarantee under no. TC210804A, Secure and Controllable System for Malicious Code Analysis under no. TC200H02X.

References

[1] A. A. Brincat, F. Pacifici, S. Martinaglia, and F. Mazzola, "The Internet of Things for Intelligent Transportation Systems in Real Smart Cities Scenarios," in *Proceedings of the 2019 IEEE*

5th World Forum on Internet of Things (WF-IoT), pp. 128–132, Limerick, Ireland, April 2019.

[2] M. Antic and I. Papp, “Smart home integration with external iot device platforms and services,” *IEEE Consumer Electronics Magazine*, vol. 10, 2020.

[3] F. Sadikin, T. v. Deursen, and S. Kumar, “A zigbee intrusion detection system for iot using secure and efficient data collection,” *Internet of Things*, vol. 12, Article ID 100306, 2020.

[4] Z. Wang, L. Feng, S. Yao, K. Xie, and Y. Chen, “Low-cost and Long-Range Node-Assisted Wifi Backscatter Communication for 5g-Enabled Iot Networks,” *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 8540457, 2021.

[5] A. J. C. Sunder and A. Shanmugam, “Jensen-shannon divergence based independent component analysis to detect and prevent black hole attacks in healthcare WSN,” *Wireless Personal Communications*, vol. 107, no. 4, pp. 1607–1623, 2019.

[6] N. A. Hikal, M. Y. Shams, H. Salem, and M. M. Eid, “Detection of black-hole attacks in manet using adaboost support vector machine,” *Journal of Intelligent and Fuzzy Systems*, vol. 41, no. 1, pp. 669–682, 2021.

[7] S. Kanthimathi and P. J. Rani, “Defending against packet dropping attacks in wireless adhoc networks using cluster based trust entropy,” in *Proceedings of the 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 2447–2452, IEEE, Bangalore, India, September 2018.

[8] B. Feng, A. Tian, S. Yu, J. Li, H. Zhou, and H. Zhang, “Efficient cache consistency management for transient iot data in content-centric networking,” *IEEE Internet of Things Journal*, 2022.

[9] B. Feng, H. Zhou, G. Li, Y. Zhang, K. Sood, and S. Yu, “Enabling machine learning with service function chaining for security enhancement at 5g edges,” *IEEE Network*, vol. 35, no. 5, pp. 196–201, 2021.

[10] A. Mitrokotsa and C. Dimitrakakis, “Intrusion detection in manet using classification algorithms: the effects of cost and model selection,” *Ad Hoc Networks*, vol. 11, no. 1, pp. 226–237, 2013.

[11] S. Marti, T. J. Giuli, K. Lai, and M. Baker, “Mitigating routing misbehavior in mobile ad hoc networks,” in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pp. 255–265, Boston Massachusetts USA, August 2000.

[12] X. Li, R. Lu, X. Liang, and X. Shen, “Side channel monitoring: packet drop attack detection in wireless ad hoc networks,” in *Proceedings of the 2011 IEEE International Conference on Communications (ICC)*, pp. 1–5, IEEE, Kyoto, Japan, July 2011.

[13] C. Pu and S. Hajjar, “Mitigating forwarding misbehaviors in rpl-based low power and lossy networks,” in *Proceedings of the 2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pp. 1–6, IEEE, NV, USA, January 2018.

[14] V. L. Narayana, A. P. Gopi, D. Anveshini, and G. V. Lakshmi, “Enhanced path finding process and reduction of packet droppings in mobile ad-hoc networks,” *International Journal of Wireless and Mobile Computing*, vol. 18, no. 4, p. 391, 2020.

[15] C. Pu, S. Lim, B. Jung, and M. Min, “Mitigating stealthy collision attack in energy harvesting motivated networks,” in *Proceedings of the MILCOM 2017-2017 IEEE Military Communications Conference (MILCOM)*, pp. 539–544, IEEE, MD, USA, October 2017.

[16] C. Pu and S. Lim, “A light-weight countermeasure to forwarding misbehavior in wireless sensor networks: design, analysis, and evaluation,” *IEEE Systems Journal*, vol. 12, no. 1, pp. 834–842, 2018.

[17] C. Pu and S. Lim, “Spy vs. spy: Camouflage-based active detection in energy harvesting motivated networks,” in *Proceedings of the MILCOM 2015-2015 IEEE Military Communications Conference*, pp. 903–908, IEEE, FL, USA, December 2015.

[18] C. Pu, S. Lim, B. Jung, and J. Chae, “Eyes: mitigating forwarding misbehavior in energy harvesting motivated networks,” *Computer Communications*, vol. 124, pp. 17–30, 2018.

[19] R. Akbani, T. Korkmaz, and G. Raju, “A machine learning based reputation system for defending against malicious node behavior,” in *Proceedings of the IEEE GLOBECOM 2008-2008 IEEE Global Telecommunications Conference*, pp. 1–5, IEEE, LA, USA, December 2008.

[20] X. Liu, M. Abdelhakim, P. Krishnamurthy, and D. Tipper, “Identifying malicious nodes in multihop iot networks using dual link technologies and unsupervised learning,” *Open Journal of Internet Of Things (OJIOT)*, vol. 4, no. 1, pp. 109–125, 2018.

[21] L. Liu, Z. Ma, and W. Meng, “Detection of multiple-mix-attack malicious nodes using perceptron-based trust in iot networks,” *Future Generation Computer Systems*, vol. 101, pp. 865–879, 2019.

[22] L. Yang, L. Liu, Z. Ma, and Y. Ding, “Detection of selective-edge packet attack based on edge reputation in iot networks,” *Computer Networks*, vol. 188, Article ID 107842, 2021.

[23] N. Tariq, M. Asim, F. A. Khan, T. Baker, U. Khalid, and A. Derhab, “A blockchain-based multi-mobile code-driven trust mechanism for detecting internal attacks in internet of things,” *Sensors*, vol. 21, no. 1, p. 23, 2020.

[24] T. Sakthivel and R. M. Chandrasekaran, “A dummy packet-based hybrid security framework for mitigating routing misbehavior in multi-hop wireless networks,” *Wireless Personal Communications*, vol. 101, no. 3, pp. 1581–1618, 2018.

[25] M. Zaminkar, F. Sarkohaki, and R. Fotohi, “A method based on encryption and node rating for securing the rpl protocol communications in the iot ecosystem,” *International Journal of Communication Systems*, vol. 34, no. 3, Article ID e4693, 2021.

[26] A. Karati, C. I. Fan, and R. H. Hsu, “Provably secure and generalized signcryption with public verifiability for secure data transmission between resource-constrained iot devices,” *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10431–10440, 2019.

[27] G. Soós, D. Ficzere, and P. Varga, “The pursuit of nb-iot transmission rate limitations by real-life network measurements,” in *Proceedings of the 2020 43rd International Conference on Telecommunications and Signal Processing (TSP)*, pp. 430–434, IEEE, Milan, Italy, July 2020.

[28] E. Nwafor, M. Robson, and H. Olufowobi, “Dynamic load sharing in memory constrained devices: a survey,” in *Proceedings of the 2021 IEEE 7th World Forum on Internet of Things (WF-IoT)*, pp. 586–591, LA, USA, June 2021.