Research Article

Design and Implementation of an Anomaly Network Traffic Detection Model Integrating Temporal and Spatial Features

Ming Li ¹, Dezhi Han ¹, Xinming Yin ^{2,3}, Han Liu ¹ and Dun Li ¹

¹College of Information, Shanghai Maritime University, Shanghai 201306, China

²The Third Research Institute of the Ministry of Public Security, Shanghai 200031, China

³Department of Computer Science and Engineering, East China University of Science and Technology, 130 Meilong Road, Shanghai 200237, China

Correspondence should be addressed to Dezhi Han; dezhihan88@sina.com

Received 26 May 2021; Accepted 7 August 2021; Published 23 August 2021

Academic Editor: Zhili Zhou

Copyright © 2021 Ming Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the rapid development and widespread application of cloud computing, cloud computing open networks and service sharing scenarios have become more complex and changeable, causing security challenges to become more severe. As an effective means of network protection, anomaly network traffic detection can detect various known attacks. However, there are also some shortcomings. Deep learning brings a new opportunity for the further development of anomaly network traffic detection. So far, the existing deep learning models cannot fully learn the temporal and spatial features of network traffic and their classification accuracy needs to be improved. To fill this gap, this paper proposes an anomaly network traffic detection model integrating temporal and spatial features (ITSN) using a three-layer parallel network structure. ITSN learns the temporal and spatial features of the traffic and fully fuses these two features through feature fusion technology to improve the accuracy of network traffic classification. On this basis, an improved method of raw traffic feature extraction is proposed, which can reduce redundant features, speed up the convergence of the network, and ease the imbalance of the datasets. The experimental results on the ISCX-IDS 2012 and CICIDS 2017 datasets show that the ITSN can improve the accuracy of anomaly network traffic detection while enhancing the robustness of the detection system and has a higher recognition rate for positive samples.

1. Introduction

In recent years, more and more enterprises choose to migrate their business to the cloud to promote the rise of cloud service providers, while the rapid development of the Internet and the strong support of national policies also promote China's cloud computing industry into the fast lane of development. The rapid development of cloud technology has made cloud security issues become more and more prominent. Some advanced cloud open network technologies have been used by criminals as a new type of criminal means which seriously affect the stable operation of cloud open network, cause national economic losses, and even threaten national security [1–5].

There are many methods for network security protection, such as authentication, access control, data encryption,

and intrusion detection [6–8]. The research in this paper belongs to the field of intrusion detection and uses a deep learning approach. Notably, the attack traffic data is essentially different from the normal traffic data. By identifying differences in the feature between the two traffic data, anomaly network traffic detection can effectively detect and intercept network attacks in advance and effectively reduce the losses caused by network attacks [9–11]. The early detection and classification methods of abnormal traffic were based on the fixed rules formulated manually for matching and classification. Later, with the emergence of machine learning methods, the model designed based on manually selected features began to be used by researchers for classification. However, the above two classification methods both require manual feature selections that are difficult in the complex and changeable network environment.

For the problems of the above traditional methods, considering that deep learning technology can independently extract data features for classification, researchers began to introduce deep learning technology into the field of anomaly network traffic detection and achieved good results. So far, deep learning still faces many challenges, such as the problem of data imbalance [12, 13]. Excessive data deviation causes the prediction results of the model to be closer to the data with high data volume, which leads to an increase in the prediction error rate. For anomaly network traffic detection, the amount of data varies greatly between different types of attacks, so the data required for modelling is equally uneven. For example, DDoS and PortScan attacks involve a large number of packets each time, while infiltration attacks involve a very small number of packets each time.

There are three commonly used methods to face the data imbalance problem, which are undersampling, oversampling, and co-sampling. Each of these three methods has its own advantages and disadvantages, and each of them has been used by a large number of scholars, but the three methods mentioned above are not used in this paper. This paper uses a unique data processing method and the feature fusion technique used in ITSAN to mitigate the impact of data imbalance on the experimental results. In the classification of traffic data, the fusion of features of multi-scale can effectively improve the classification accuracy and reduce the impact of data imbalance on the experimental results. Feature fusion is divided into early fusion and late fusion, and early fusion is used in this paper.

The raw data processing method proposed in this paper is described in detail as follows. Firstly, network traffic data are divided into traffic according to the source IP address, source port, destination IP address, destination port, and transport layer. Secondly, the traffic data is extracted. The data distribution of the raw traffic data is retained in the extraction process. The raw data are used to perform feature learning, and the traffic data with insufficient features is filled forward. This method avoids the introduction of too many zero elements, increases the robustness of the model, and can speed up the convergence of the model.

Since most current deep learning methods only consider the features of network traffic in terms of temporal or spatial features, it is difficult to fully extract the inherent features of network traffic data, which directly leads to a decrease in the detection accuracy. This paper proposes an anomaly network traffic detection model fusing temporal and spatial features (ITSAN). ITSAN combines deep learning technology, feature fusion technology, and the features of the traffic data, which can automatically fully learn the intrinsic features of traffic data (including spatial and temporal features) [14, 15].

The contribution of this paper can be summarized as follows:

- (1) A new temporal and spatial feature fusion model (ITSAN) is proposed. ITSAN can not only fully learn the spatial and temporal features of traffic data but also effectively solve the data imbalance problem through multiple feature fusion.
- (2) A new data extraction method is proposed, including setting an upper limit for the number of packets

contained in each traffic data to avoid the problem of losing a large amount of packet information when intercepting some data from traffic and sending them to the network for training.

- (3) The experimental results on the ISCX-IDS 2012 [16] and CICIDS 2017 [17] datasets show that the ITSAN can improve the accuracy of anomaly network traffic detection while enhancing the robustness of the detection system and has a higher recognition rate for positive samples.

The remainder of this paper is organized as follows. Section 2 reviews the relevant work of the methods in the field of anomaly network traffic detection. Section 3 introduces the proposed ITSAN and the improved raw traffic extraction method. In Section 4, we conduct ablation experiments on the proposed ITSAN using the ISCX-IDS 2012 and CICIDS 2017 datasets. Finally, the paper is summarized in Section 5.

2. Related Works

This section introduces some related research studies in the field of anomaly network traffic detection, including traditional anomaly network traffic detection methods and anomaly network traffic detection methods based on machine learning and deep learning.

2.1. Traditional Methods. This section briefly describes some traditional anomaly network traffic detection methods. Zhou et al. [18] proposed a prediction-based method for analysing anomalous behaviours. The experimental results show that the method has high accuracy but weak detection capability. To solve the problem of weak detection ability of traditional methods, Jiang et al. [19] proposed a new method to detect traffic anomalies in the network by accurately capturing features in the transition domain. For the problem of too high data dimensions, Juvonen et al. [20] proposed a framework to find abnormal behaviour from these logs. Although traditional anomalous traffic detection algorithms can achieve relatively good detection results, most of the traditional algorithms have high computational complexity, require specific priori knowledge, and are weak in handling encrypted traffic.

2.2. Methods Based on Machine Learning. Machine learning-based anomaly network traffic detection methods are the most researched methods. Kilincer et al. [21] provided a detailed review of datasets commonly used in the field of anomaly network traffic detection and examined traditional machine learning methods compared to each other. To further improve the detection accuracy and the processing time of the algorithm on the dataset, Agarwal et al. [22] proposed an integrated approach that integrates multiple machine learning methods to improve detection accuracy. Because most machine learning-based anomaly network traffic detection methods nowadays rarely consider data quality, Gu and Lu [23] proposed an effective intrusion

detection framework-based SVM with plain Bayesian feature embedding. Machine learning-based methods for anomalous network traffic detection are excellent, but there is a general problem with machine learning-based methods, that is, they cannot automatically learn traffic features, so there is a need to design a set of features that can accurately reflect the traffic characteristics, for which researchers have started to introduce deep learning into the field.

2.3. Methods Based on Deep Learning. In 2006, Professor Hinton [24] proposed the concepts of pretraining technology and deep learning. In 2012, deep learning technology shined in the image classification competition and a research boom is set off in academia and industry. Since then, the research in the field of image has become hot, and it also promotes the research in the field of intrusion detection [25–27]. Zhang et al. [28] proposed an anomaly network traffic detection method based on deep-level network and raw traffic data providing a detailed raw data processing process. Zhang et al. [29] proposed a deep learning model (PCCN) with two layers of parallel learning and cross fusion, and the experimental results were good. Hwang et al. [30] proposed an effective anomaly network traffic detection mechanism (D-PACK), which consists of a convolutional neural network (CNN) [24] and an unsupervised deep learning model (autoencoder) to automatically describe traffic pattern and filter anomaly traffic. High detection accuracy can still be achieved even when only the first two packets in traffic data are intercepted. Yan and Xu [31] designed a new hinge classification algorithm based on mini-batch gradient descent with an adaptive learning rate and momentum (HCA-MBGDALRM) to minimize the effects of security attacks.

For two common problems in the field of network anomaly detection (feature dependence and high false-positive rate), Wei and Wang [32] proposed a feature learning (HAST-NAD) network anomaly detection method with hierarchical temporal and spatial features based on deep learning. This method can automatically learn traffic features and improve the efficiency of anomaly network traffic detection. Kim and Cho [33] proposed a C-LSTM neural network that can effectively model the temporal and spatial feature information contained in traffic data and automatically extract temporal and spatial feature information from the raw data.

With the rapid development of the network, the types of anomaly network traffic tend to become more complicated, and the network model trained only by specific data sets is difficult to be applied in the real-time changing network environment. As a result, some researchers have begun to improve the network model in a deeper level, allowing them to identify those unknown attacks not in the training set. Table 1 summarizes the research results of deep learning in the field of anomaly network traffic detection mentioned in this section. Alzawairi et al. [34] proposed two new classification methods for unknown attacks. To solve the problems of anomaly network detection and improve accuracy and scalability, Khan et al. [35] proposed a new network

model based on spark ML and convolutional LSTM. Zhang et al. [36] proposed an open set classification network (OCN) method to detect unknown attacks using the nearest class average (NCM) classifier and designed two new loss functions (Fisher loss and maximum average difference (MMD) loss) to jointly optimize the loss.

3. Model and Methods

3.1. Structure of ITSN. As shown in Figure 1, ITSN consists of two parallel convolutional network layers and a spatial-temporal layer and uses the idea of feature fusion to fuse spatial and temporal features. Besides, the ITSN can improve the learning performance of traffic data features and effectively solve the data imbalance problem by combining feature fusion techniques. Feature fusion can achieve the complementary advantages between different features, which is more conducive to fully learning the intrinsic features of traffic data to weaken the impact of data imbalance on traffic data feature learning [37]. As shown in Figure 2, we replace the long short-term memory (LSTM) network with a convolutional module to verify the importance of learning temporal features for traffic feature learning.

Part of the design of ITSN was inspired by paper [38]. ITSN has a three-layer structure. The first layer and the third layer learn the spatial features of the traffic from different aspects. The second layer simultaneously learns spatial and temporal features of the traffic, and the features of each layer are fused several times during the learning process of the model to strengthen the result of learning.

The first layer of the network structure which is fully convolved is inspired by the FCN [39] in the field of image segmentation. FCN can achieve pixel-level semantic segmentation, so using a network structure similar to FCN can get more detailed traffic characteristics. Because a large number of information features will be lost in the pooling layer, the pooling layer is replaced by a convolution layer, which can learn more fully the spatial features without losing the traffic data information.

In the second layer, it is considered that the traffic data features learned by using only the convolutional network will lose the temporal features of the traffic data. In this layer, we only use convolutional neural network (CNN) and long short-term memory network (LSTM), but no other structures are used, CNN is used to extract the spatial features of the traffic data, LSTM is used to extract the temporal features of the traffic data, and the combination of the two structures can extract the intrinsic temporal and spatial features of the traffic data.

This layer is mainly formed by the convolutional layers and pooling layers stacked on top of each other, in which the pooling layer can remove the redundant information and expand the receptive field. Therefore, the features learned by the network will be more adequate after adding the pooling layer.

The three-layer model proposed in this paper is linked by feature fusion. Fusing features of multi-scale are an important means to improve classification performance, to fully learn traffic data features and to effectively reduce data imbalance.

ITSN adopts three feature fusion techniques and all adopt the “concat” fusion method [14]. Through multiple

TABLE 1: Some research results of deep learning in the field of anomaly network traffic detection.

Method	Datasets	Year	Reference
PCCN (parallel cross convolutional neural network)	CICIDS 2017	2019	[29]
D-PACK	USTC-TFC2016	2020	[30]
HAST-NAD (CNN + RNN)	ISCX2012	2020	[32]
C-LSTM	ISCX2012	2018	[33]
Shallow, deep ANN	UNSW-NB15	2020	[34]
Conv-LSTM	ISCX-IDS 2012	2019	[35]
A scalable network intrusion detection system	KDDCUP'99 CICIDS 2017	2021	[36]

FIGURE 1: Network structure of ITSN.

FIGURE 2: Network structure of ISN.

feature fusion, the learning of traffic features is enhanced to reduce the impact of data imbalance on model performance and improve classification accuracy.

3.2. Features of ITSN. In ITSN, the features learned by the multi-layer network are fused many times, the time-series feature learning module LSTM is introduced, and the time-series features and spatial features are fused before the feature learning.

LSTM is a very important and classic model in the field of natural language processing. It consists of a forget gate, an input gate, and an output gate. The forget gate is used to selectively forget information. The input gate is responsible

for recording new information into the state, and the output gate determines the final output value. Under the combination of these three gates, the time-series features of the traffic data can be learned well.

The advancement of the ITSN model structure lies in the following:

- (1) The spatial features learned by CNN are sent to the LSTM module for time-series feature learning.
- (2) The learned time-series features are fused with the spatial features learned in the first and second layer, and then the features of the mixed features are extracted for classification.

The formula of neural network from input to output (contains only one hidden layer) is shown in the following formulas:

$$a_h = \sum_{i=1}^n w_{ih} b_i, \quad (1)$$

$$b_h = f(a_h). \quad (2)$$

Change the dimensionality of the output data of the CNN network, as shown in the following formula:

$$x_1 = \text{reshape}(b_h). \quad (3)$$

The calculation formula of the input gate of LSTM is shown in the following formula:

$$b_l^1 = f\left(\sum_{i=1}^L w_{il} x_i^1 + \sum_{h=1}^L w_{hl} b_h^0 + \sum_{c=1}^n w_{cl} s_c^0\right). \quad (4)$$

The calculation formula of the forget gate of LSTM is shown in the following formula:

$$b_{\varnothing}^t = f\left(\sum_{i=1}^L w_{i\varnothing} x_i^1 + \sum_{h=1}^L w_{h\varnothing} b_h^0 + \sum_{i=1}^C w_{c\varnothing} s_c^0\right). \quad (5)$$

The calculation formula of the LSTM cell module is expressed as follows:

$$s_c^1 = b_l^1 g\left(\sum_{i=1}^L w_{ic} x_i^1 + \sum_{h=1}^L w_{hc} b_h^0\right) + b_{\varnothing}^1 s_c^0. \quad (6)$$

The calculation formula of the LSTM output gate is expressed as follows:

$$\varnothing_c^0 = f\left(\sum_{i=1}^L w_{i\varnothing} x_i^t + \sum_{h=1}^L w_{h\varnothing} b_h^h + \sum_{c=1}^C w_{hw} s_c^0\right) h(s_c^0). \quad (7)$$

The formula for fusion of CNN output and LSTM output is expressed as follows:

$$\text{torch.cat}(x_{\text{first}}, x_{\text{third}}, \text{reshape}(b_c^0)). \quad (8)$$

In the CNN formula, w_{1h} represents the parameters between the first layer and the second layer, where b_i represents the input data, f represents the activation function, and $\text{reshape}()$ is the function of transforming the data dimension.

In the LSTM formula, x_i^t represents the normal input, b_l^h represents the previous moment the hidden layer input, s_c^1 represents the previous state stored in the cell, w_{il} , w_{hl} , and w_{cl} are all the connection weights between the layers of the LSTM structure, and f represents the activation function.

3.3. The Methods of Data Processing. The raw traffic data in the pcap file is stored in binary format and cannot be fed directly into the neural network for training. It needs to be processed first. Effective data processing can enhance the feature information of the traffic data and remove the

interference information, thus improving the accuracy of the experiment.

To reduce the impact of data imbalance in the traffic datasets, we propose a raw traffic feature extraction method. References [28, 29] put forward an effective data processing method to eliminate a large number of zero data that are useless for feature learning. This paper inherits and improves the methods provided by Zhang et al. [28, 29]. Firstly, the upper limit of the number of packets contained in each traffic is set according to the five-tuple partition flow, so that the number of traffic data contained in an attack traffic increases, which is equivalent to using a data enhancement method to alleviate data imbalance (there are many methods for data enhancement, one of which is to copy data to increase the amount of data that was originally small). Secondly, a specific number of packets are intercepted in a traffic data. If the number of packets is less than the intercepted number, the forward filling method is adopted, which is conducive to the feature learning of short and small traffic data. The original data processing method proposed in this paper not only reduces the interference of useless data to the experimental results but also reduces the artificially added 0 data by using the antecedent filling method, which further reduces the interference and is more conducive to the extraction of features by the neural network especially for the extraction of short data stream features. The specific steps of the data processing method are as follows.

Step 1: firstly, compare the label file with the pcap file. Secondly, separate the abnormal network traffic data from the pcap file. Thirdly, store the abnormal network traffic data into the csv file named after the attack type. To summarize, it is to separate the anomalous type of traffic from the pcap file provided by the dataset.

The pseudocode of Algorithm 1 is as follows.

Step 2: firstly, read the attack traffic data from the csv file. Secondly, divide it into traffic using five-tuple. Thirdly, set a threshold for the number of data packets contained in each traffic when the data packets are divided into traffic. Finally, use the `hexdume()` function to convert the binary traffic data into hexadecimal data and store it in a txt file. To summarize, it converts byte type data into hexadecimal data and divides the packets into flows for storage. The pseudocode of Algorithm 2 is as follows.

Step 3: read the data stored in the txt file and save the read data as a matrix. The pseudocode of Algorithm 3 is as follows.

Step 4: set size and length to extract specific data from data []. If the number of data packets contained in a traffic is smaller than the size, the specific data are filled by forward filling. If the number of bytes of each data packet in a traffic is smaller than the length, the data are filled with zero. To summarize, this step is the last step of data processing for flow slicing, and after this step, the data can be sent to the network for training. The pseudocode of Algorithm 4 is as follows.

```

Input: network traffic pcap files.
Output: completed data of the raw traffic and their labels.
For each pcap do
  Create seven null lists srcip = [], dstip = [], sport = [], proto = [], raw data = [], labels = [].
  If the five-tuple information of traffic packet could be found in the attack labels
    Trim source IP into the list srcip.
    Trim destination IP into the list dstip.
    Trim source port into the list sport.
    Trim destination port into the list dport.
    Trim protocol into the list proto.
    Get packet data into the list raw data.
    Get label into the list labels.
  Add above lists to the csv file, and the file name is named based on the date.
END if
END for

```

ALGORITHM 1: Raw traffic data extraction.

```

Input: csv file containing raw traffic
Output: a txt file with hexadecimal data converted from the raw traffic data
For  $i = 0$  to data_excel.length do
  IF determine whether the data packet has been divided into streams
    Count = 0
    For  $j = i + 1$  to data_excel.length do
      If 5-tuple equal
        Count+ = 1
        If Count > threshold:
          Change the file name of the stream
          Count = 0
        END if
      Open a file
      Use the hexdump function to convert the data packet into hexadecimal and write it into the same file
      Mark that the packet has been divided into traffics
    END if
  END for
END for

```

ALGORITHM 2: Get numerical data.

4. Experiment and Result Analysis

In this paper, three sets of ablation experiments were designed to validate the performance and robustness of ITSN. Experiment 1 used the ISCX-IDS 2012 ID dataset and Experiment 2 used the CICIDS 2017 dataset, respectively, for ablation experiments on the ITSN. Experiment 3 sets up an exploratory experiment. In this group of experiments, we have studied how intercepting data from traffic to form the traffic attributes fed to the CNN for training can give best results.

4.1. Experimental Environment. The experimental environment is shown in Table 2.

4.2. Datasets. At present, there are many datasets used for anomaly network traffic detection. The KDD99, an early dataset used in scientific research, and NSL-KDD, a dataset that improves on KDD99, both of which were widely used in the early years and are still used by many scholars to evaluate

intrusion detection methods. However, both NSL-KDD and KDD99 are 20-year-old datasets that do not take into account new network attacks. They are difficult to train anomaly network traffic detection models that are adapted to the current complex network environment. For this reason, this paper uses ISCX-IDS 2012 and CICIDS 2017 datasets, which have more stability and robustness than NSL-KDD and KDD99 in training network models [40].

The ISCX-IDS 2012 dataset is composed of dynamic network performance data, which contains 7 days of normal and anomaly network traffic activities. Two of the days have no attack traffic and the other five days contain a total of four attack scenarios (internal penetration of the network HTTP, denial of service, distributed denial of service using the IRC botnet, and brute force SSH attacks) [16]. CICIDS 2017 is a dataset for intrusion detection and intrusion prevention, which was open-sourced by the Canadian Institute of Cyber Security in 2017. The dataset provides csv and pcap files, which contain traffic data collected from Monday to Friday [17].

```

Input: raw txt file
Output: data []
File = Open ()
For line in file.readlines() do
  Set Flag = 0
  For i = 3 to 19 do
    Flag = 1
    If determine whether a data packet is over:
      Flag = 0
      Break
    END if
  END for
  Store the package data in the intermediate list mid_data []
  If Flag = 0:
    Write mid_data[] data to the list data[]
    mid_data = []
  END if
END for

```

ALGORITHM 3: Get the feature matrix.

```

Input: data [], size, length
Output: the final data used for neural network training
For i = 0 to data.length do
  If data[i] ≥ size
    Count = 1
    For j = 0 to size do
      For k = 0 to length:
        If k greater than or equal to data[i][j].length
          Write data to 0 into the .csv file
          Count + 1
        Else:
          Write data[i][j] to the .csv file
        END if
      END for
    END for
  Else:
    Count = 1
  END for

```

ALGORITHM 4: Get the final data that can be fed into the neural network for training.

In the ISCX-IDS2012 dataset and the CICIDS 2017 dataset, some traffic characteristics are very similar, while some traffic characteristics are quite different. We extracted data from the ISCX-IDS 2012 dataset and visualized four types of attack samples using the data processing method proposed in Section 3. The results are shown in Figure 3. From the picture, we can clearly see that there are some types of traffic features that are more distinct and some that are less distinct.

4.3. Evaluation Metrics. This section focuses on the criteria for evaluating the merits of deep learning methods and machine learning methods in the field of anomaly network traffic detection. In the evaluation criteria, all evaluation indicators are obtained based on the two-dimensional confusion matrix of the actual class and the predicted class [41]. The confusion matrix is shown in Table 3.

The diagonal of the confusion matrix indicates the correct prediction and the nondiagonal elements are the wrong predictions of a certain classifier. According to the confusion matrix, several commonly used evaluation indicators can be obtained as follows.

Precision: it is defined as the sample with the correct prediction divided by all samples, as shown in the following formula:

$$\text{precision} = \frac{TP}{TP + FP}. \quad (9)$$

Recall rate: it represents the ratio of the number of samples that are correctly classified as a certain type to the actual number of samples, as shown in the following formula:

TABLE 2: Experimental environment of this paper.

CPU	Intel(R)Core (TM)i7-10875H CPU@2.30 GHz
GPU	RTX 2060
RAM	16G
OS	Windows 10 Professional Edition
Compiler environment	Python 3.8.2

FIGURE 3: A visualization of the features of different types of attack traffic. (a) Infiltrating. (b) HTTP. (c) Distributed denial. (d) Bruteforce SSH.

TABLE 3: Confusion matrix.

Actual class	Attack Normal	Predicted class	
		Attack	Normal
		True positive (TP) False positive (FP)	False negative (FN) True negative (TN)

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (10)$$

FPR: it is also known as the false-positive rate, which is defined as all correctly predicted samples divided by all normal samples, as shown in the following formula:

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}. \quad (11)$$

Accuracy: it represents correctly predicted samples divided by all samples. It is also called detection accuracy.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \quad (12)$$

F1_score: it is the summed average of the precision and recall rates, with a maximum of 1 and a minimum of 0, as shown in the following formula:

$$\text{F1}_{\text{score}} = 2 * \left(\frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \right). \quad (13)$$

The above criteria for evaluating model quality, precision, recall, and accuracy are all closer to 1 indicating a better model, while the false-positive rate is closer to 0 indicating a better model.

4.4. Hyperparameter Settings. In ITSN, all convolution kernels are of $3 * 3$ size because too large convolutional kernels will lead to excessive loss of traffic data information, and too small convolutional kernels will increase the computational effort and affect the real-time performance of model detection. In the model training process, we use the

Adam optimizer to speed up the convergence of the network, with the momentum fixed at 0.9 and the weight decay set to 5×10^{-4} to prevent overfitting. Different learning rates have different effects on the convergence speed. In a total of 15 epochs, the learning rate of the first 8 rounds is set to 10^{-3} , the learning rate of the last 3 rounds is 10^{-4} , and the learning rate of the last two rounds is set to 10^{-5} . All experiments are performed on an RTX2060 graphics card and the batch size is set to 256. In the verification and testing phase, this paper sets the batch size to 512. To effectively verify the performance of the model, we did not use additional data enhancement techniques in the training phase and testing phase.

4.5. Experiment and Result. To verify the performance of the ITSN in anomaly network traffic detection, we compared the ITSN with the classic abnormal network traffic detection model on the data sets ISCX-IDS 2012 and CICIDS 2017. The experimental results show that the ITSN model is significantly better than others.

Figures 4(a)–4(e) show the experimental results of the model on the ISCX-IDS 2012 dataset. The overall detection accuracy of the model is the highest as seen in Figure 4(e), indicating that ITSN is better than other comparable models. It can be seen from Figures 4(b) and 4(c) that the value of F1_score and the value of recall of the ITSN are both optimal. The optimal F1_score indicates that the ITSN is more robust than other models, and the optimal recall indicates that the ITSN has a stronger ability to recognize positive samples than other models. The overall performance of the ITSN in the test is better than that of the ISN, which shows the importance of learning the time-series features of traffic data.

FIGURE 4: Comparison of experimental results. (a) Precision. (b) Recall. (c) F1_score. (d) FPR. (e) Average accuracy.

TABLE 4: Precision comparison

Label	0	1	2	3	4	5	6	7	8	9	10	11
BILSTM	0.9976	1	0.9817	0.9942	0.9707	0.9933	0.9997	1	0.9935	0.9999	0.9996	0.9962
ISN	1	1	0.9925	0.9993	0.9853	0.9957	1	1	1	1	0.9996	1
PCCN	1	1	0.9863	0.9993	0.9901	0.9762	1	1	1	0.9999	1	0.9986
CNN	1	0.9999	0.946	0.9979	0.9719	0.989	0.9997	0.9985	0.9981	0.9999	0.9996	0.9962
CNN_LSTM	0.9952	0.9999	0.9692	0.9912	0.9727	0.9669	1	0.9995	0.9991	0.9999	0.9995	0.9943
ITSN	1	1	0.9941	0.9993	0.9896	0.9924	1	1	1	1	0.9998	0.9957

TABLE 5: Recall comparison.

Label	0	1	2	3	4	5	6	7	8	9	10	11
BILSTM	0.9831	1	0.8637	0.9996	0.975	0.9877	0.999	1	0.9981	0.9999	0.9989	0.9967
IST	1	1	0.9925	0.9993	0.9853	0.9957	1	1	1	1	0.9996	1
PCCN	1	1	0.9825	0.9995	0.9566	0.9938	1	1	1	1	0.9996	1
CNN	0.9928	1	0.9506	0.9979	0.9669	0.9839	0.999	1	1	0.9999	0.9989	0.9967
CNN_LSTM	0.9952	1	0.7958	0.9994	0.9183	0.9843	0.998	1	0.9981	0.9997	0.9993	0.9957
ITSN	1	1	0.9827	0.9997	0.9838	0.9943	1	1	1	1	1	1

TABLE 6: F1_score comparison.

Label	0	1	2	3	4	5	6	7	8	9	10	11
BILSTM	0.9903	1	0.919	0.9969	0.9728	0.9905	0.9994	1	0.9958	0.9999	0.9993	0.9964
ISN	1	1	0.9874	0.9995	0.9896	0.9941	0.9999	1	1	1	0.9998	0.9993
PCCN	1	1	0.9844	0.9994	0.973	0.985	1	1	1	0.9999	0.9998	0.9993
CNN	0.9964	1	0.9483	0.9979	0.9694	0.9864	0.9994	0.9992	0.9991	0.9999	0.9993	0.9964
CNN_LSTM	0.9952	1	0.874	0.9953	0.9447	0.9756	0.999	0.9997	0.9986	0.9998	0.9994	0.995
ITSN	1	1	0.9884	0.9995	0.9867	0.9934	1	1	1	1	0.9999	0.9979

TABLE 7: FPR comparison.

Label	0	1	2	3	4	5	6	7	8	9	10	11
BLSTM	0	0	0.0003	0.004	0.0002	0.0001	0	0	0	0	0	0
ISN	0	0	0.0003	0.0002	0	0.0001	0	0	0	0	0	0
PCCN	0	0	0.0002	0.0002	0.0001	0	0	0	0	0	0	0
CNN	0	0	0.001	0.0014	0.0002	0.0001	0	0	0	0	0	0
CNN_LSTM	0	0	0.0005	0.0061	0.0002	0.0003	0	0	0	0	0	0.0001
ITSN	0	0	0.0001	0.0005	0.0001	0.0001	0	0	0	0	0	0

TABLE 8: Overall accuracy comparison.

Label	ITSN	ISN	PCCN	CNN	CNN_LSTM	BILSTM
Accuracy	0.9994	0.9993	0.9992	0.9978	0.9954	0.9978

To further verify the effectiveness of the ITSN, this paper also did the same experiment on the dataset CICIDS 2017 and plotted the test results in Tables 4–8 (the optimal results have been marked in bold in the tables). The ITSN still maintains the highest detection accuracy on the CICIDS 2017 dataset, and it also has the best F1_score and recall as the test results on the ISCX-IDS 2012 dataset, indicating that ITSN is versatile and stable. To better show the misclassification of the traffic data of the ITSN on the CICIDS 2017 dataset, we have drawn the testing result into a heat map as shown in Figure 5 (the number on the diagonal represents the number of correct predictions of the model), and it can be seen that most of the traffic data of each category can be

correctly classified, and the reason for a small number of classification errors may be that its characteristics are not obvious or the similarity with other traffic characteristics is high.

From the results of the two ablation experiments, it can be seen that the proposed ITSN not only can get better experimental results than other models but also has universality and stability. The performance of ISN is slightly worse than that of ITSN model because the ISN model ignores the learning of traffic data time-series features.

From the above two ablation experiments, it can be concluded that the detection accuracy of the proposed

FIGURE 5: Thermal diagram of test results of ITSN model.

FIGURE 6: Box plot of experimental results.

model on the dataset with relatively higher complexity CICIDS 2017 is higher than that of ISCX-IDS 2012. As mentioned above, it is possible that some of the data in the ISCX-IDS 2012 dataset were incorrectly labelled, but this may not be the whole reason. We found that the amount of data contained in each type of traffic in CICIDS 2017 is larger than that in ISCX-IDS 2012, and deep learning is data-oriented, so the more the data, the more the features extracted and learned and the better the classification effect. Overall, the above two reasons

ultimately lead to the proposed model in this paper performing better on the more complex CICIDS 2017 dataset instead.

4.6. Exploratory Experiment. The reasons for the traffic data extraction include the following:

- (1) Traffic data contains too much information that cannot all be fed to the network for training.

TABLE 9: Comparison of overall accuracy of test experiment.

Package number	20	40	60	80	100	120	140	160	180
1	0.93	0.9927	0.9933	0.9946	0.9947	0.9949	0.995	0.995	0.995
2	0.9496	0.9939	0.9952	0.995	0.9953	0.728	0.9945	0.9963	0.9962
3	0.9531	0.6257	0.9954	0.9953	0.9955	0.9953	0.9961	0.996	0.9961
4	0.9596	0.9963	0.9963	0.9966	0.9964	0.9967	0.9963	0.9962	0.9956
5	0.9605	0.9963	0.9954	0.9957	0.9958	0.996	0.9956	0.996	0.9961
6	0.9611	0.9968	0.9968	0.9967	0.9963	0.9966	0.9952	0.9954	0.9949

FIGURE 7: Continued.

FIGURE 7: Visualized line chart of experimental data. (a) Accuracy. (b) Precision. (c) Recall. (d) F1_score. (e) FPR.

TABLE 10: Test results.

Model	Accuracy	Precision	Recall	F1_score	False-positive rate
5(96)	0.9988	0.9983	0.99845	0.9983	0.00045
4(100)	0.9988	0.9982	0.998325	0.9983	0.000475

- (2) The deep learning model requires the data which are input to the network to have the same data size.
- (3) There are a lot of data in traffic data which are disturbing and useless for feature learning.

The traffic data extraction method of the first two ablation experiments in this section is fixed, each traffic intercepts the first 5 data packets, and each data packet intercepts the first 96 bytes. But we have questions about whether this interception method is the optimal interception method, so we did the exploratory experiment. In this exploratory experiment, we use the ISCX-IDS 2012 dataset, and the model we use for convenience is the simple CNN model.

Considering that the maximum number of data packets in the traffic is 6 by using the raw traffic extraction method proposed by us, when we study the influence of the number of data packets intercepted from traffic on the experimental results, the number of packets we intercepted is between 1 and 6. Most packets contain no more than 180 bytes, so the number of bytes intercepted per packet is an arithmetic sequence with a tolerance of 20, with values ranging from 20 to 80, as shown in the following formula:

$$S_n = na_1 + \frac{n(n-1)}{2}d, \quad n \in N^*. \quad (14)$$

Box plots allow us to visualize the anomalies (i.e., outliers) in the experimental data, so we plotted the experimental results as a box plot as shown in Figure 6 (the points marked with the coordinates in the box plot are the outlier points).

Table 9 shows that there are many outliers in the experimental data (already marked in bold in the table), and it is obvious from Table 9 that when the length of bytes intercepted by each packet is 20 bytes, all the experimental results are all judged as outliers, indicating that the effect of feature learning by intercepting bytes of that length is obviously inferior to other interception lengths. To obtain more accurate results, we removed the most obvious outliers (0.728 and 0.6257) marked in the table when plotting the line graph shown in Figure 6.

As shown in Table 9, the test set accuracy is higher when 4, 5, and 6 packets are intercepted in each traffic data, and a good test set accuracy is obtained as long as the byte length of each packet interception is greater than 40 bytes. When the byte length of each packet intercepted exceeds 80 bytes, the accuracy of the experiment basically starts to stabilize. To facilitate observation, we plot the experimental results as a line graph shown in Figure 7. It can be seen more intuitively from Figure 7 that no matter how many data packets are intercepted from each traffic data, the experimental results start to stabilize when the length of each packet exceeds 80 bytes.

We tested the ITSN model using the ISCX-IDS 2012 dataset by intercepting the first 4 packets from each traffic data and the first 100 bytes of each packet, and the test results are shown in Table 10.

Through this exploration experiment, it can be seen that the data interception method of the first two comparative ablation experiments was not optimal and there was room for improvement. As can be seen in Table 11, the interception method obtained by this exploratory experiment can

TABLE 11: Time consumption.

Cut method	Loading time (s)	Training time (s)	Testing time (s)
5(96)	6.79	566.65	3.33
4(100)	5.64	541.12	2.83

improve the real-time performance of the ITSN without reducing the experimental accuracy of the ITSN.

5. Summary and Outlook

Security issues are becoming more and more prominent, although encryption is a very good security measures [41] but there are shortcomings, the combination of encryption and intrusion detection in network protection will have better results. Because the current anomaly network traffic detection methods cannot fully learn the spatial and temporal features of the traffic, this paper proposes an anomaly network traffic detection model ITSN, which is composed of three layers. The three-level features are fused by feature fusion technology, and the fused features are fed into global convolution to extract features for classification. We conducted ablation experiments on the ITSN using ISCX-IDS 2012 and CICIDS 2017 datasets, and the experimental results show that the ITSN can not only effectively improve the detection accuracy of anomaly network traffic but also maintain strong robustness and positive sample recognition rate. Besides, to reduce the impact of data imbalance on the experimental results, this paper proposes a new method to extract features from the raw traffic. This method can not only effectively reduce the impact of data imbalance on the whole model but also reduce redundant features and speed up the convergence of the network.

Although the ITSN is superior in all aspects, the implementation of the ITSN is based on a closed set protocol. Under the closed set protocol which means that the training set of the training network is fixed and does not change, the classification process only considers the known classes in the training and does not detect the unknown attacks or even misclassifies them as the known classes in the training so the scalability of the ITSN needs to be improved. Considering the increasingly complex network environment, the extensible open set recognition model can be better applied in the actual network environment, so in future research, we will further study and solve the ITSN scalability problem. Then, ITSN can be deployed to a real-time network environment for testing and refinement.

Data Availability

The datasets used in this study can be accessed from the following link: <https://www.unb.ca/cic/datasets/ids-2017.html>. The processed datasets can be availed by contacting the author at 3030310118@stu.shmtu.edu.cn.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This research was supported by the National Natural Science Foundation of China under grant nos. 61873160 and 61672338 and Natural Science Foundation of Shanghai under grant no. 21ZR1426500. In addition to the authors listed in the paper, Ming-Ming Cui, Shao-Kang Cai, and Zhen-Hui Wang also contributed to the content studied in this paper and are gratefully acknowledged.

References

- [1] M. Cui, D. Han, and J. Wang, "An efficient and safe road condition monitoring authentication scheme based on fog computing," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 9076–9084, 2019.
- [2] M. Ahmed, A. Naser Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, 2016.
- [3] M. Cui, D. Han, J. Wang, K.-C. Li, and C.-C. Chang, "ARFV: an efficient shared data auditing scheme supporting revocation for fog-assisted vehicular ad-hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 15815–15827, 2020.
- [4] W. Wang, *Deep Learning for Network Traffic Classification and Anomaly Detection*, Hei Fei: University of Science and Technology of China, Hefei, China, 2018.
- [5] T. Xiao, D. Han, J. He, K.-C. Li, R. F. de Mello, and R. Fernandes, "Multi-Keyword ranked search based on mapping set matching in cloud ciphertext storage system," *Connection Science*, vol. 33, no. 1, pp. 95–112, 2021.
- [6] W. Liang, S. Xie, D. Zhang, X. Li, and K. Li, "A mutual security authentication method for RFID-PUF circuit based on deep learning," *ACM Transactions on Internet Technology*, pp. 1–20, 2020.
- [7] J. Su, R. Xu, S. Yu, B. Wang, and J. Wang, "Redundant rule detection for software-defined networking," *KSII Transactions on Internet and Information Systems*, vol. 14, no. 6, pp. 2735–2751, 2020.
- [8] W. Liang, D. Zhang, X. Lei, M. Tang, K.-C. Li, and A. Zomaya, "Circuit copyright blockchain: blockchain-based homomorphic encryption for IP circuit protection," *IEEE Transactions on Emerging Topics in Computing*, vol. 9, p. 1, 2020.
- [9] J. Jia, J. Yang, and L. I. Tao, "A DPI-based autocorrelation method for packet detection classification," *Journal of Sichuan University (Natural Science Edition)*, vol. 56, no. 1, pp. 29–36, 2019.
- [10] C. Orlikowski, R. Hein, and E. Wittbrodt, "Port-based approach to distributed transfer function method," *Proceedings in Applied Mathematics and Mechanics*, vol. 9, no. 1, pp. 671–672, 2009.
- [11] Q. Tian, D. Han, K.-C. Li, X. Liu, L. Duan, and A. Castiglione, "An intrusion detection approach based on improved deep belief network," *Applied Intelligence*, vol. 50, no. 10, pp. 3162–3178, 2020.
- [12] O. Kemal, C. C. Baris, K. Sinan et al., "Imbalance problems in object detection: a review," p. 1, 2020, <https://arxiv.org/abs/1909.00169>.

- [13] H. Haibo He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.
- [14] Y. Dai, F. Gieseke, S. Oehmcke, Y. Wu, and K. Barnard, "Attentional feature fusion," in *Proceedings of the 2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 3559–3568, Waikoloa, HI, USA, January 2021.
- [15] K. M. He, X. Y. Zhang, and S. Q. Ren, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2014.
- [16] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Computers & Security*, vol. 31, no. 3, pp. 357–374, 2012.
- [17] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proceedings of the International Conference on Information Systems Security & Privacy*, pp. 108–116, 2018.
- [18] Y. Zhou, G. Hu, and D. Wu, "A data mining system for distributed abnormal event detection in backbone networks," *Security and Communication Networks*, vol. 7, no. 5, pp. 904–913, 2014.
- [19] D. Jiang, Z. Xu, P. Z. Zhang, and T. Zhu, "A transform domain-based anomaly detection approach to network-wide traffic," *Journal of Network and Computer Applications*, vol. 40, pp. 292–306, 2014.
- [20] A. Juvonen, T. Sipola, and T. Hmlinen, "Online anomaly detection using dimensionality reduction techniques for HTTP log analysis," *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 91, pp. 46–56, 2015.
- [21] I. F. Kilincer, F. Ertam, and A. Sengur, "Machine learning methods for cyber security intrusion detection: datasets and comparative study," *Computer Networks*, vol. 188, Article ID 107840, 2021.
- [22] A. Agarwal, P. Sharma, M. Alshehri et al., "Classification model for accuracy and intrusion detection using machine learning approach," *PeerJ Computer Science*, vol. 7, no. 3, p. e437, 2021.
- [23] J. Gu and S. Lu, "An effective intrusion detection approach using SVM with naïve Bayes feature embedding," *Computers & Security*, vol. 103, Article ID 102158, 2021.
- [24] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, no. 2, 2012.
- [25] Z. L. Zhou, Q. M. Jonathan Wu, Y. M. Yang, and X. M. Sun, "Region-level visual consistency verification for large-scale partial-duplicate image search," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 16, no. 2, pp. 1–25, 2020.
- [26] Z. L. Zhou, Y. Mu, and Q. M. Jonathan Wu, "Coverless image steganography using partial-duplicate image retrieval," *Soft Computing*, vol. 23, no. 13, pp. 4927–4938, 2019.
- [27] W. Liang, J. Long, K.-C. Li, J. Xu, N. Ma, and X. Lei, "A fast defogging image recognition algorithm based on bilateral hybrid filtering," *ACM Trans. Multimedia Comput.*, vol. 17, no. 2, p. 42, 2021.
- [28] Y. Zhang, X. Chen, L. Jin et al., "Network intrusion detection: based on deep hierarchical network and original flow data," *IEEE Access*, vol. 7, pp. 37004–37016, 2019.
- [29] Y. Zhang, X. Chen, D. Guo et al., "PCCN: parallel cross convolutional neural network for abnormal network traffic flows detection in multi-class imbalanced network traffic flows," *IEEE Access*, vol. 7, no. 99, p. 1, 2019.
- [30] R. H. Hwang, M. C. Peng, C. W. Huang et al., "An unsupervised deep learning model for early network traffic anomaly detection," *IEEE Access*, vol. 8, pp. 30387–30399, 2020.
- [31] X. Yan and Y. Xu, "Trustworthy network anomaly detection based on an adaptive learning rate and momentum in IoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6182–6192, 2020.
- [32] G. Wei and Z. Wang, "Adoption and realization of deep learning in network traffic anomaly detection device design," *Soft Computing*, vol. 25, no. 3, 2020.
- [33] 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.
- [34] M. Alzewairi, S. Almajali, and M. Ayyash, "Unknown security attack detection using shallow and deep ANN classifiers," *Electronics*, vol. 9, no. 12, Article ID 2006, 2020.
- [35] M. Khan, R. Karim, and Y. Kim, "A scalable and hybrid intrusion detection system based on the convolutional-LSTM network," *Symmetry*, vol. 11, no. 4, p. 583, 2019.
- [36] Z. Zhang, Y. Zhang, D. Guo et al., "A scalable network intrusion detection system towards detecting, discovering, and learning unknown attacks," *International Journal of Machine Learning and Cybernetics*, vol. 12, pp. 1–17, 2021.
- [37] W. Liang, L. Xiao, K. Zhang, M. Tang, D. He, and K. C. Li, "Data fusion approach for collaborative anomaly intrusion detection in blockchain-based systems," *IEEE Internet of Things Journal*, p. 1, 2021.
- [38] J. Su, R. Xu, S. Yu, B. Wang, and J. Wang, "Idle slots skipped mechanism based tag identification algorithm with enhanced collision detection," *KSII Transactions on Internet and Information Systems*, vol. 14, no. 5, pp. 2294–2309, 2020.
- [39] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, 2017.
- [40] Z. Ahmad, A. S. Khan, C. W. Shiang et al., "Network intrusion detection system: a systematic study of machine learning and deep learning approaches," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 1, 2020.
- [41] D. Z. Han, N. N. Pan, and K.-C. Li, "A traceable and revocable ciphertext-policy attribute-based encryption scheme based on privacy protection," *IEEE Transactions on Dependable and Secure Computing*, no. 99, p. 1, 2020.