This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2933165, IEEE Access

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

PCCN: Parallel Cross Convolutional Neural Network for Abnormal Network Traffic Flows Detection in Multi-class imbalanced Network Traffic Flows

YONG ZHANG, XU CHEN, DA GUO, MEI SONG, YINGLEI TENG, XIAOJUAN WANG
School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China

Corresponding author: XU CHEN(buptchenxu@gmail.com)

This work is supported by the State Major Science and Technology Special Projects (2014ZX03004002)

ABSTRACT Network attack behavior detection using deep learning is an important research topic in the field of network security. Currently, there are still many challenges in detecting multi-class imbalanced abnormal traffic data. This paper proposed a new intrusion detection network based on deep learning, named parallel cross convolutional neural network (PCCN), to improve the detection performance of imbalanced abnormal flows. By fusing the flow features learned from the two branch convolutional neural networks (CNN), PCCN can better learn the flow features with fewer samples, to improve the detection results of the imbalanced abnormal flows. We proposed an improved feature extraction method of the original flow to extract multi-class flow features at the same time. The proposed algorithm not only reduces the number of useless elements for network learning, but also accelerates network convergence. In addition, we proposed four improved versions of the PCCN network structure to meet the real-time requirements of network intrusion detection in the current big data computing. These networks can achieve almost the same detection results as the PCCN, but greatly reduce the detection time of data. Through the analysis of high-order evaluation metrics, the proposed PCCN algorithm is significantly better than the traditional machine learning algorithms. Compared with the current hierarchical network model, PCCN can also achieve better performance in term of overall accuracy.

INDEX TERMS Network Intrusion Detection, Cross Network, Deep Learning, Feature Fusion

I. INTRODUCTION

In recent years, with the rapid development of the Internet and the increase in the number of users, the network security and management have suffered great threats. Lawbreakers use various methods of attack to launch attacks on the network, which not only causes huge economic losses but also brings privacy problems to users. Network security has been concerned by many organizations. People hope to detect attacks in time through effective tools or systems and take corresponding measures to reduce the losses. Intrusion detection technology is an effective attack detection method. The attack behavior can be detected by analyzing the traffic data generated by network users. The key challenge is how to effectively identify traffic data with aggressive behaviors. At present, many researchers have made great efforts to improve the classification and recognition of abnormal traffic. However, there are several problems with previous work. First, researchers pay too much attention to the overall accuracy of abnormal traffic classification and ignore the classification accuracy of imbalanced samples. Second, the datasets used by researchers are relatively old, with fewer diverseness and quantities. Third, the researchers do not consider the efficiency of the algorithm in the context of big data.

Imbalanced data learning [1] has always been a serious problem in the field of machine learning, and the problem of data imbalance seriously affects the performance of algorithms [2]. In the field of network intrusion detection, the data imbalance problem also exists. Specifically, the amount of traffic attack data varies greatly among different categories. For example, samples of PortScan [3] and DoS [4] attacks are relatively easy to collect, and a large number of data samples can be obtained, while data samples similar to APT

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2933165, IEEE Access

[5] attacks are difficult to collect. In order to effectively detect abnormal traffic, the traffic data is usually divided into flow [6] according to the five-tuple information. The abnormal traffic with attack behavior is the most commonly used method by detecting and analyzing the flow. The data classification and recognition for imbalanced network traffic mainly include the methods based on feature selection [7], the methods based on data resampling [8] and the methods based on cost sensitive learning [9].

Feature selection-based methods usually select a subset of features with better classification performance from flow data, so that redundant features can be removed and those flow features with stronger distinguishing ability can be retained. However, feature selection methods are mainly used in the data preprocessing steps of machine learnings, and usually do not consider the correlation between features and categories [10]. In addition, the feature selection methods have a serious problem that the selected feature subset is usually biased towards the category with a large number of abnormal traffic.

Data resampling is the most commonly used method to deal with imbalanced data classification. It is an effective method originally used to detect imbalanced abnormal flow data. However, due to the variety of network attack categories, massive traffic data will be generated every day, so the method based on data resampling is no longer suitable for detecting imbalanced abnormal traffic data. There are two serious drawbacks to data resampling: One drawback is that the data resampling will change the original distribution of data by introducing new sample instances. And producing a new sample or changing the original distribution by adding new samples or instances might add noise to the dataset which will lead to further problems. In contrast, our proposed method will not generate new data instances and thus eliminating the problem associated with noise data. A better result will be presented in the “ABLATION STUDIES” section. Furthermore, data resampling is often very time-consuming, and the kind of algorithm with high time complexity cannot meet the requirements of the current big data computing. In “Analysis of Test time” section, we gave the SMOTE resampling method time in the dataset (about 7.8h).

The cost-sensitive method learns better features by calculating a cost matrix for misclassified samples. Due to the large difference in the number of different samples of imbalanced abnormal traffic data, the weight of various samples can be learned through cost sensitivity. It has been shown in many fields [11] [12] that the cost-sensitive learning method is superior to the sampling-based method in dealing with imbalanced data sample classification. In the field of imbalanced data learning, the algorithm combining cost-sensitive learning and deep neural network [2] [13] has gradually become an effective alternative to the data resampling method. Network traffic data is gradually encapsulated and composed according to the network protocols, and each layer of the protocols have fixed field information. Usually, researchers will extract various fields from the flow data to effectively detect traffic data with abnormal behaviors. However, there is no uniform standard for how to extract features and what kind of features to extract. Blake Anderson et al. [14] extracts several hundred-dimensional features from flow data for malware traffic recognition, while M Lopez-martin et al. [15] only extracts six dimensional flow features, which can achieve 96% accuracy in abnormal flow detection. However, for the detection of imbalanced abnormal traffic data, manually designed features cannot achieve a better detection effect for categories with small data volume. In view of the above problems, we propose the following principles to the problem of imbalanced abnormal traffic data classification:

1) Preserve the original feature distribution of flows, remove the complex manual feature design process and use the original data of flow to perform feature learning.

2) We propose an improved flow original feature extraction algorithm. This algorithm avoids introducing too many 0 elements that are not helpful for network learning, and can speed up the convergence of the network.

3) The detect network model should have low test time cost to meet the requirements of big data computing.

In this paper, we propose a new parallel cross convolutional neural network(PCCN) based on feature fusion, which uses the original features of flows to improve the detection results of imbalanced abnormal traffic data. The experimental results show that the proposed method can effectively improve the detection performance of imbalanced abnormal flow data and only cause a little detection delay. Code has been released at https://github.com/chenxu93/abnormal-traffic. The main contributions of this paper are as follows:

(1) We propose a new network model, named PCCN, to improve the detection performance of highly imbalanced abnormal flow through feature fusion.

(2) We propose an improved flow original feature extraction algorithm. This algorithm avoids introducing too many 0 elements that are not helpful for network learning, and can speed up the convergence of the network.

(3) For our proposed new network model, we propose four improved versions to reduce detection time and meet the requirements of attacks detection in big data computing.

The remainder of this paper is organized as follows. The second section reviews the relevant work of the imbalanced data detection methods in network intrusion detection. The third section describes in detail the network model used in this paper to solve highly imbalanced abnormal flow data. In section four, we conduct ablation experiments on the CICIDS2017 dataset to evaluate the effectiveness of our proposed model. Finally, we summarize our paper in the fifth section.

II. RELATED WORKS

Although the method of imbalanced data classification learning has attracted extensive attention in academia and industry [1] [13], it has not received sufficient attention in the field of intrusion detection. Considering a large amount of traffic data, the general method to solve the imbalanced data

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

classification is difficult to be used to solve the imbalanced abnormal traffic data classification detection. At present, most methods for dealing with imbalanced abnormal traffic detection are mainly based on the data level and combined with traditional machine learning algorithms to improve the detection performance of multi-class imbalanced abnormal traffic data.

Parsaei M R et al. [16] proposed a joint method of SMOTE, cluster center and nearest neighbor, and conducted experiments on the NSL KDD dataset, achieving the classification accuracy of 94% and 50% for U2R and R2L attack samples respectively. Although classification accuracy is improved, the joint method proposed by them is relatively complex, and only accuracy, detection rate and false alarm are considered. Yan B H et al. [17] proposed a SMOTE and feature selection method to improve the detection rates of U2R and R2L in the NSL KDD dataset. Since SMOTE and feature selection will change the original distribution of the data set and bring additional time consumption, this does not suitable for network intrusion detection in big data environments. Shi H et al. [18] considered that high-dimensional traffic data has redundant features, which affect the detection performance of imbalanced abnormal traffic data. Therefore, they proposed a feature generation and feature selection method based on deep learning to improve the detection performance of imbalanced abnormal flow. Shen J et al. [19] proposed a method of second feature extraction and feature sampling to reduce the dimensions of imbalanced network traffic data. They use the multidimensional assessment algorithm to perform second feature extraction on the sub-datasets. The experimental results show the effectiveness of their method for the detection of imbalanced abnormal traffic data on the NSL KDD dataset. Peng L et al. [20] improved from the model and proposed a new gravitation-based classification model. They constructed six imbalanced flow sub-datasets using the original traffic dataset. The experimental results show that the gravitation-based classification model has higher accuracy than the traditional machine learning algorithm for the classification of imbalanced traffic data. Bamakan S M H et al. [20] improved the loss function and proposed the Ramp-KSVCR model. This model can solve highly imbalanced attack data and requires less training time, which is suitable for large-scale datasets. The classification false alarm rate of highly imbalanced abnormal traffic data is reduced on NSL KDD and unsw-nb15 [21] dataset. Liu Z et al. [22] proposed a new class-oriented feature selection (COFS) method and adopts ensemble learning mechanism to enhance the generalization ability of abnormal traffic data detection. Their method can effectively overcome the negative impact of data drift on the classifier, and achieves more than 96% flow classification accuracy and more than 93% average byte classification accuracy on real-world network traffic data. Cui Z et al. [23] proposed a bat algorithm and used deep learning methods to detect imbalanced malicious code. By converting malicious codes into grayscale images and sending them to a neural network for automatic recognition, high detection accuracy of malicious codes can be achieved and the detection speed is very fast.

However, there are some problems with the above methods. First, the researchers focus on the final overall classification metrics, while ignoring the classification metrics of each type of abnormal traffic in the dataset. Second, features of KDD and NSL KDD datasets are fixed and feature dimensions are not high. It is unreasonable to select features on these non-original distributed datasets. Third, many algorithms do not consider time cost, although they can improve the detection of imbalanced abnormal traffic data but can not meet the requirements of big data computing.

In this paper, we improve the original flow data feature extraction method [24] and reduces the 0 elements that are useless for learning, which not only reduces the time for feature extraction, but also greatly reduces the storage space. Then we propose a new parallel cross convolutional neural network (PCCN) to improve the evaluation metrics of imbalanced abnormal traffic data classification in intrusion detection. Finally, in order to consider the time efficiency of the algorithm, we propose some improved versions of the PCCN, which can balance the detection efficiency and time efficiency. In the experimental part, we pay more attention to improving each category of evaluation metrics of imbalanced abnormal flow data, rather than the overall evaluation metrics. Compared with our previous work [24], the proposed algorithm has a lower test time with more data. In particular, we compared the methods proposed by Abdulhammed R [25] in the experiment section. Similarly, on the CICIDS2017 dataset, they adopted Features Dimensionality Reduction Approaches to improve the over accuracy of imbalanced categories, but our method achieves better experimental results.

III. SYSTEM MODEL

In this section, we design a new network named PCCN to improve the detection performance of imbalanced abnormal traffic data. PCCN mainly consists of two parallel convolutional neural networks, and uses the idea of feature fusion to make the network pay more attention to the categories with fewer data samples. Using the original features of flows, the network can automatically learn and improve the classification metrics of imbalanced abnormal traffic. In order to enable the network to use traffic data for training, we need to convert each flow into a two-dimensional grayscale diagram. The flow generation process will be first described below.

A. DATA PREPROCESSING

This paper inherits the method of using original flow data in [24] [26] and improves the method of preprocessing previous data, so that the proposed method can better express the features of flows. Based on the five-tuple information of the traffic packets, we divide them into flows. According to our statistical analysis, we find that the number of traffic packets in each flow is from three to several thousand. However, most flows contain only five traffic packets, so we limit the number

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

```
Algorithm 1: original flow data extraction
Input: network traffic pcap files.
Output: Header, Payload, Header&Payload features of the original flows and theirs labels.
Step 1: transform pcap files to txt files.
for each pcap do
    Get flows based on the five-tuple information of traffic packages.
    for each flow do
        Transform flow pcap file into txt file with tshark to get flow's original hexadecimal data
    end
end
Step 2:Extract header, payload, header&payload features of the original flows from txts
create three null lists flows=[], flows_payload=[], flows_head_payload=[].
for each txt file do
    initialize three lists to store three categories of original flow features, flow_feature=[256], payload_feature=[256], header_payload_feature=[].
    for each package do
        1, get 1st to 50th hexadecimal bytes to generate header features.
        2, get 51st to 100th hexadecimal bytes to generate payload features.
        3, get 1st to 96th hexadecimal bytes to generate header&payload features.
        if the number of bytes in the package less than the target feature do:
            fill to the target feature with 0
        if package >= 5 do
            label the three types feature base on five-tuple information, then add to last dimension
            update flow_feature, payload_feature and header_payload_feature vectors
            add above vectors to flows, flows_payload and flows_head_payload respectively
        else do:
            fill to 5 packages with 0.
    end
end
```

of traffic packets in a timestamp to five. The specific flow feature extraction process is as follows:

(1) The traffic packets are divided into flows according to the five-tuple information, which is different from the previous method [24]. Considering the real-world network environment, we reserve the header fields of MAC layer. In addition, each flow uses five traffic packets. There are two advantages to doing this. On the one hand, it reduces redundant features, which mainly come from the top layer. On the other hand, the features of the flows are more compact. Although the previous method in [24] reduces the 0 element in the flows features as much as possible, the features of flows are still very sparse, which has no positive effect on network learning.

(2) For the flows of more than 5 traffic packets, we take the extra traffic packet as a new flow. Because the traffic packets with the same five-tuple do not regard the timestamp information as a flow feature, the division method considering the timestamp is too inaccurate. Because the time span of traffic packets in a flow is very large, it is unreasonable to ignore the timestamp information.

(3) In the flow features, we use the hexadecimal number 'FF' to distinguish each traffic packet, instead of the number 0. We use the number 0 to fill in the insufficient features of flows and the number of traffic packets that are missing in a flow (assuming at least three traffic packets in a flow).

The new flow features generation algorithm has the following advantages. First, more flow samples can be obtained from the original dataset. Second, avoid introducing too many 0 elements, and the features of the flow become more compact. Third, the Header, Payload and Header&Payload features of the flow can be extracted simultaneously. The original feature extraction method of flows is shown in algorithm 1.

B. MODEL DESCRIPTION

In order to improve the classification metrics of imbalanced abnormal traffic data, we adopt CNN for feature learning. CNN has good spatial perception ability and good feature learning ability for image-related tasks. Many CNN-based feature extraction networks are developed and validated for computer vision tasks. In this paper, PCCN is proposed to improve the classification metrics of multi-class imbalanced abnormal traffic data. The PCCN mainly includes two branch networks for feature learning which is shown in Figure 1. The proposed model of PCCN combines features extracted from the two branch networks, in which the top branch uses fully convolutional networks (FCN) and bottom branch uses ordinary CNN. The feature fusion by the two branches makes the features of network learning more distinguishable and robust. Below we detail the PCCN model.

1) Top Branch

The basic idea of PCCN is mainly inspired by the semantic segmentation algorithm FCN [27]. FCN is first used to handle image segmentation tasks that can handle pixel-level classification. Generally, the sample number of highly imbalanced traffic data is less than its feature number. So in order to make the algorithm better learn the features with fewer samples, the top branch network of parallel cross convolutional neural network adopts FCN. Pooling layers are abandoned in our FCN, the key of which is to use convolution operation instead of pooling operation for down-sampling. Using FCN can bring two gains. First, there are more convolution layers in FCN, so that richer semantic information can be obtained from flow data. Second, FCN can be used to flexibly control network parameters, so that the network will not be too complex to prevent the risk of overfitting.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

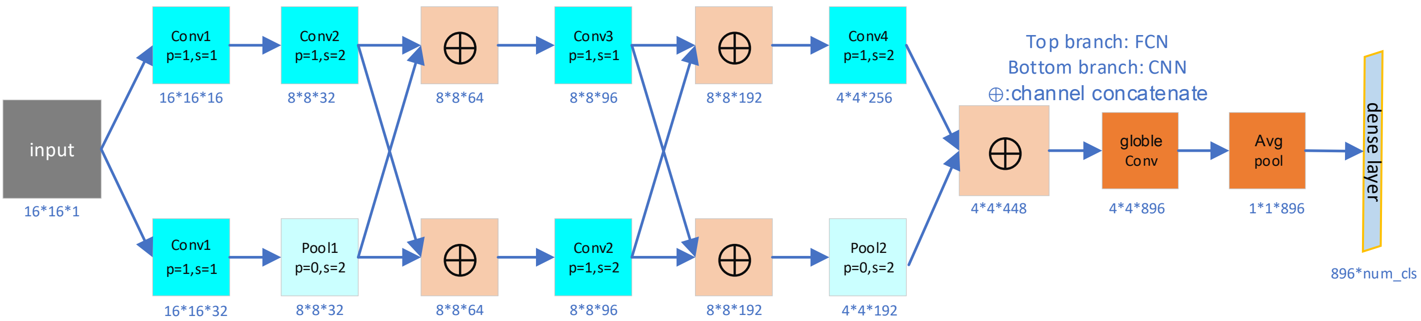


FIGURE 1. Parallel Cross Convolutional Neural Network

2) Bottom Branch

For imbalanced abnormal traffic data, we need to consider that the algorithm should not to learn too many redundancy features. Since the convolution kernel has a large number of overlapping computational regions on the feature map when doing the sliding convolution, we should avoid the network learning too many redundant features. A large number of redundant features make the model biased toward samples with large data volume during detection, thus reducing the detection metrics of abnormal flow data with few samples. Pooling operation reduces redundant features by calculating values in a small region, then it outputs only one value as a feature. Because each traffic packet in a flow has different information, attack traffic of the same category often contains very little information. We need to consider the invariance of different traffic packets in the same flow, while the pooling layer has the translation, rotation, and scale invariance. Therefore, our second branch adopts conventional CNN and uses pooling layer for down-sampling. In addition, pooling layer is adopted for down-sampling, which can not only reduce the size of the feature map but also reduce the parameters of network learning.

3) Feature Cross Fusion

In the field of computer vision, many networks are proposed to solve the classification detection performance of small objects or blurry objects. These networks are mainly improved from the feature level to make network learning more expressive features. Feature pyramid networks (FPN) [28] combines the rich semantic features of the top layer with the high-resolution features of the lower layers to improve the detection results of small objects. In order to further enhance the feature learning of flow with less sample size, the feature cross fusion is performed in PCCN.

Specifically, our proposed PCCN performs three feature fusion operations. First, after the first down-sampling of the two branch networks, the output feature maps adopt channel cascade. The channel cascade operation does not change the size of the feature map, only doubles the number of channels. Second, the first fused feature maps are passed through a 3*3 sliding convolution window respectively, and then the output feature maps perform channel cascade again. After the second channel cascade operation, we perform down-sampling operations on the two branch networks respectively to reduce the size of the feature maps. Finally, we perform channel cascade operation on feature maps of the second down-sampling of the two branches, and then fuse the features of each channel through a global convolution operation and further increase the nonlinearity of the network. After three times of feature fusion, we adopt a global average pooling layer to reduce the size of feature map and reduce the redundancy of features, so as to prevent the features learned from the model being biased to the abnormal flow categories with large sample size. The output of the network uses the dense layer and the softmax layer to perform multi-class imbalanced abnormal flow classification detection. In addition, a batch normalization (BN) layer is added after each convolutional layer, and all activation functions use ReLU.

4) Convolution and Batch Normalize

In the PCCN, all convolution kernels are 3*3 size, because the smaller convolution kernel can effectively reduce the complexity of the model. In addition, the stacking of multi-layer 3*3 convolution kernels can also obtain the equivalent receptive field of the larger convolution kernel. For the convolution operation, assuming that χ_{i-1} is the input feature map of the current layer and the input size of the convolution kernel ω_i is 3*3, then the output feature map χ_{i-1} of the current layer is expressed as:

$$\chi_i{=}\omega_i * \chi_{i-1} + b_i \tag{1}$$

Where b_i denotes the bias term. Due to the large difference in the distribution of flow data samples of different categories, for highly imbalanced abnormal flow data, the network adapts well to the abnormal flow categories with large sample size, while the detection effect is often poor for flows with few samples. To prevent overfitting, the batch normalization (BN) is introduced [29]. The batch normalization accelerates the convergence of the training depth network model by reducing the internal covariate shift of the input data. For imbalanced data, the introduction of batch normalization reduces the correlation of each feature dimension and enables the network to fit the categories with small sample size as quickly as possible. For an input mini-batch, the data

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

is normalized by calculating the mean and variance of the samples in this mini-batch. The process of Batch normalize transformation [29] is as follows:

For m input samples, the mean μ_B and variance σ_B^2 of the mini-batch are calculated respectively, and then normalized according to the mean and variance. The specific process is as follows:

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{2}$$

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_B)^2 \tag{3}$$

$$\widehat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} \tag{4}$$

μ_B represents the mean of the feature dimensions of all samples in a mini-batch, σ_B^2 representing the variance of all samples in the mini-batch. The sample x_i is normalized by μ_B and σ_B^2 to obtain $\widehat{x}_i$. Each input layer uses the normalize operation to make all input data get the same distribution without sample diversity. In order to make the network learning more robust, the features are scaled and shifted by introducing two learnable parameters and enforce the model learn the original samples distribution to enhance the generalization of different data.

$$BN_{\gamma,\beta}(x_i) = \gamma x_i + \beta \tag{5}$$

After the data perform batch normalize, the nonlinearity is introduced through an activation function. In our network structure, in order to speed up the convergence of the network and solve the effects of gradient explosion and gradient disappearance, all activation functions use the ReLU function. So the final output feature map is expressed as:

$$z_i = g(BN_{\gamma,\beta}(\omega_i * \chi_{i-1} + b_i)) \tag{6}$$

Where g denotes the activation function. Finally, softmax function is used for our classification, and the sample output of each category is expressed as:

$$y_i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}} \tag{7}$$

IV. EXPERIMENT ANALYSIS

In this section, we conduct a large number of ablation studies of the proposed PCCN model. The effectiveness of our proposed method is illustrated by a large number of ablation experiments. All of our experiments are conducted on CICIDS2017 dataset [30]. Our experimental environment is as follows:

CPU: Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz
GPU: 2*GTX1080ti 11GB
RAM: 32GB
OS: Ubuntu 16.04

TABLE 1. The distribution of 12 categories attack flows

Label	Flow type	Number	Percentage
0	BotNet	2075	0.18%
1	DDoS	261226	22.35%
2	Goldeneye	20543	1.76%
3	Dos Hulk	474656	40.62%
4	Dos Slowhttp	6786	0.58%
5	Dos Slowloris	10537	0.90%
6	FTP Patator	19941	1.71%
7	HeartBleed	9859	0.84%
8	Infiltration	5330	0.46%
9	PortScan	319636	27.35%
10	SSH Patator	27545	2.36%
11	Web Attack	10537	0.90%

A. DATASET

Although there are many data sets available for research in the field of intrusion detection, most of the datasets lack species diversity and sample size, and some of them are quite old and different from the current attack categories. In addition, many datasets contain only header information but lack of payload information, which can not reflect the current attack trend well. A new dataset named CICIDS2017 [30] is used in this paper, which can better meet the current attack trend.

The CICIDS2017 dataset is an open source network intrusion detection and prevention dataset collected by the Canadian network security research institute in 2017. Iman Sharafaldin et al. [30] extracted 84-dimensional flow features with the CICFlowMeter-V3 tool and made accurate and reliable labeling. The dataset provides CSV files and pcap files containing flow data collected from Monday through Friday. Statistical analysis shows that there are 14 attack categories in total, but we combine Web Attack Brute Force, Web Attack XSS and Web Attack SQL Injection into Web Attack category. The reason is that they are both Web Attack categories and the number of SQL Injection and XSS sample is very small.

Using Algorithm 1, we extract twelve categories of attack flows from CICIDS2017 dataset. Compared with the previous work [24], the new original flow feature extraction method not only saves time and space costs, but also obtains more attack samples. The distribution of all attack flows in our experiment is shown in Table 1.

We extract a total of 1168,671 flow data from CICIDS2017 dataset, including 12 categories of attacks. According to the statistical results in table 1, we can find that DDoS, DoS Hulk and PortScan attacks account for a large percentage in the data set. The number of attack samples varies greatly, and the number of attack samples of Dos Hulk and BotNet differs by about 229 times. To evaluate our model performance, we randomly selected 80% from the CICIDS2017 dataset as the training set and the remaining 20% as the test set. Specifically, we do not divide them from the entire dataset, but divide them by the number of each category. The purpose of this is to ensure that the abnormal traffic categories with fewer samples can also appear in the test set, so that the

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2933165, IEEE Access

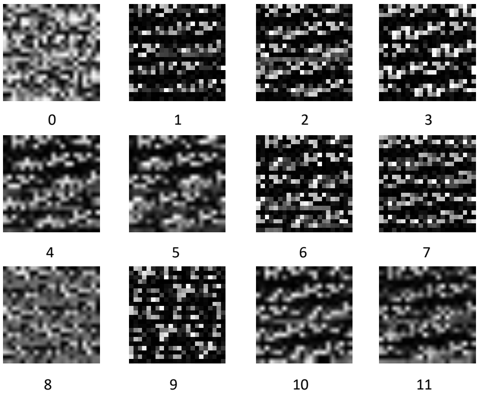


FIGURE 2. 12 categories of attack flow samples

classification metrics of each category of the sample can be reasonably analyzed.

Using our improved original flow feature extraction algorithm, we visualize 12 categories of attack flow samples, and the result is shown in figure 2. According to the visualization result, it can be found that the features of some flows are obviously different, while the features of some flows are similar.

B. EVALUATION METRICS

In order to analyze the classification detection performance of our proposed model for each category of abnormal flows, our classification metrics are similar to [7] [31], mainly to investigate the accuracy, precision, recall, and F1-measure. For a given category c_i, the following four metrics can be calculated:

True Positive (TP_{c_i}): The model label as category c_i, and the actual label is also category c_i.

False Positive (FP_{c_i}): The model label as category c_i, but the actual label is not category c_i.

True Negative (TN_{c_i}): The model label not as category c_i, and the actual label is not category c_i.

False Negative (FN_{c_i}): The model label as category c_i, but the actual label is not category c_i.

Based on the above definitions, we use the most effective and commonly used metrics [32] [33] for the evaluation of imbalanced abnormal flow data in intrusion detection, and define several high-order metrics to measure the proposed model.

$$OA = \frac{\sum\limits_{c_i=1}^{N} (TP_{c_i} + TN_{c_i})}{\sum\limits_{c_i=1}^{N} (TP_{c_i} + FP_{c_i} + TN_{c_i} + FN_{c_i})} \tag{8}$$

$$EA_{c_i} = DR_{c_i} = ER_{c_i} = \frac{TP_{c_i}}{TP_{c_i} + FN_{c_i}} \tag{9}$$

$$EP_{c_i} = \frac{TP_{c_i}}{TP_{c_i} + FP_{c_i}} \tag{10}$$

TABLE 2. The used symbols and notations

OA	the overall effectiveness of the classifier
EA_{c_i}	the accuracy rate of category c_i
DR_{c_i}	the detection rate of category c_i
ER_{c_i}	the recall rate of category c_i
EP_{c_i}	the precision rate of category c_i
EF_{c_i}	the F1-Measure rate of category c_i
EPR_{c_i}	the precision-recall of category c_i
AA	the average accuracy of the classifier
AP	the average precision of the classifier
AF	the average F1-Measure of the classifier
$G-Mean$	the G-Mean metric c_i
APR	the average precision-recall of the classifier
N	the number of imbalanced abnormal flow categories

$$EF_{c_i} = \frac{2 * EP_{c_i} * ER_{c_i}}{EP_{c_i} + ER_{c_i}} \tag{11}$$

$$EPR_{c_i} = EP_{c_i} * ER_{c_i} \tag{12}$$

$$AA = AR = \frac{1}{N} \sum_{c_i=1}^{N} EA_{c_i} \tag{13}$$

$$AP = \frac{1}{N} \sum_{c_i=1}^{N} EP_{c_i} \tag{14}$$

$$AF = \frac{1}{N} \sum_{c_i=1}^{N} EF_{c_i} \tag{15}$$

$$G-Mean = \left(\prod_{c_i=1}^{N} ER_{c_i} \right)^{1/N} \tag{16}$$

$$APR = \frac{1}{N} \sum_{c_i=1}^{N} EP_{c_i} * ER_{c_i} \tag{17}$$

The used symbols and notations are summarized in Table 2.

Overall accuracy (OA) represents the overall effectiveness of the classifier, which is the most concerned evaluation metric of most researchers. For the imbalanced abnormal flow samples in this paper, we hope to pay more attention to the accuracy of each category (EA_{c_i}), that is, the detection rate of each category of abnormal flow, which is also called the recall rate (ER_{c_i}) of each category. Similarly, we propose the precision rate (EP_{c_i}) and F1-Measure (EF_{c_i}) for each category of abnormal flows. Specifically, we evaluated the overall detection results of the model on the dataset by using a weighted average of each metric of each category of samples to obtain their corresponding high-order metrics. Here, AA denotes average accuracy, which is also equal to AR and ADR respectively. AR denotes average recall and ADR denotes average detection rate. AP denotes average precision, and AF denotes average F1-Measure. The advantage of this is that we can effectively evaluate the detection effect of our model on each category of abnormal flows.

Usually, when the model can obtain better overall detection metric, it does not mean that it has good detection effect for each category. If the model classifies the test samples

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

into one category with a large number of samples, it often gets good overall evaluation metrics. In addition, in order to more reasonably represent the detection performance of our proposed model for imbalanced abnormal flow data, we adopt the G-Mean metric. G-mean is a geometric mean that measures the recall rate for each category separately, and is calculated as shown in Eq. 16. We borrow the concept from the idea of calculating mAP in object detection tasks in computer vision. To evaluate the precision and recall simultaneously, a new metric PR is introduced, which represents the product of the precision and recall weighted average of each category of abnormal flow in the dataset. In the experiment N is 12, which represent 12 different abnormal flow categories.

C. ABLATION STUDIES

In order to study the detection performance and generalization ability of our model for imbalanced abnormal flow data, we compared the performance of the model with other common neural network models. In order to investigate the generalization capability of the model for the features we extracted, we experiment with the three types features of flow header, payload and header&payload to verify the contributions of different features to the detection effect. In addition, we compare the performance of traditional machine learning algorithms. Finally, we evaluate the test time performance of these models.

1) Experiments Details

First, we show the implementation details of the proposed network. We train all networks in an end-to-end manner. We take three features to validate our proposed model, namely, header, payload and header&payload features. For the header and payload features, we extract 256-dimensional features from each flow and then resize them into a 16*16-scale grayscale image and sent it to the network for training. Adam is used as an optimizer to speed up the convergence, and the weight decay is set to 0.0005 to prevent overfitting. For the first 5 epoches, the learning rate is set to 0.0001. For the next 3 epoches, we decay it to 0.00001, and for the last 2 epoches, we decay it to 0.000001. All experiments are performed on two 1080ti GPUs with the batch size set to 256.

In the test phase, we set the batch size to 512. It is worth noting that in order to effectively verify the performance of our proposed model, we do not use additional data augmentations during the testing phase and the training phase.

2) Analysis of PCCN

We use PCCN as the baseline and compare the performance with other network models. First, we compare the performance of PCCN and the existing hierarchical network model. In addition, we compare the experimental performance of PCCN and its improved version. Specifically, we compared the commonly used network models in the style of CNN [34] [35] and LSTM [36] [37]. Then we compared the popular hierarchical network models of the joint CNN and LSTM networks [24] [26] [38].

Considering the model complexity, four improved versions of the PCCN model are proposed. The first improved version is the Parallel Convolutional Neural Network (PCN). The feature fusion is performed only once, that is, the features are extracted separately by FCN and CNN, and then conduct feature fusion on the output of the two branch networks. Channel cascade operation is still adopted for the fusion of the two output features. The network structure is shown in figure 3(a). The second improved version is the parallel cross convolutional neural network with point convolution (PPCCN). Three feature fusions are performed, considering that the number of channels will double after each feature fusion, we use a set of 1*1 convolution kernels for channel compression. In addition, the 1*1 convolution kernel also has two functions: First, cross-channel information fusion. Second, increase the nonlinearity of the network. The PPCCN network structure is shown in figure 3(b). The third improved version is the parallel cross convolutional neural network with dilated convolution (DPCCN), which has the same network structure as PPCCN, but we put the top branch of the third convolution layer, the two convolution layers of the bottom branch and the third feature fusion layer are replaced with dilated convolution. The reason is that the dilated convolution can obtain a larger receptive field without introducing additional parameters, and it has a strong perception of background information and small objects [39]. For imbalanced anomalous flow data, we want to use dilated convolution to make the model do not learn the information about the redundant features. The fourth improved version is the parallel cross convolutional neural network by element sum (APCCN), whose network structure is the same as that of the PCCN, except that in each feature fusion, the corresponding element sum operation is adopted instead of the channel cascade operation. The use of element sum does not double the network's channels and can effectively reduce the parameters of the model. In addition, since the number of channels output by the bottom branch is half that of top branch channels, we need to unify the final number of channels. In this paper, we fill the number of channels missing in the bottom branch output feature map with 0 to fill the same number of channels as the top branch. The APCCN network structure is shown in Figure 3(c).

According to the header features of 12 categories of imbalanced abnormal flows extracted by algorithm 1, we perform experiments on each network model and calculate the DR_{c_i}, EP_{c_i}, EF_{c_i} and EPR_{c_i} of each category of flows. Based on these metrics, we further calculate the corresponding high-order metrics to measure the performance of the model.

Through the analysis of the experimental results in table 3 to table 6, we can find that there are significant differences in the four metrics of DR_{c_i}, EP_{c_i}, EF_{c_i} and EPR_{c_i} for different categories of abnormal flows. For category 3, 5, and 12 abnormal flows, the detection effect of each model is worse than other categories. For other abnormal flow categories with small samples, the model can still achieve considerable detection performance. In addition, we expect

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

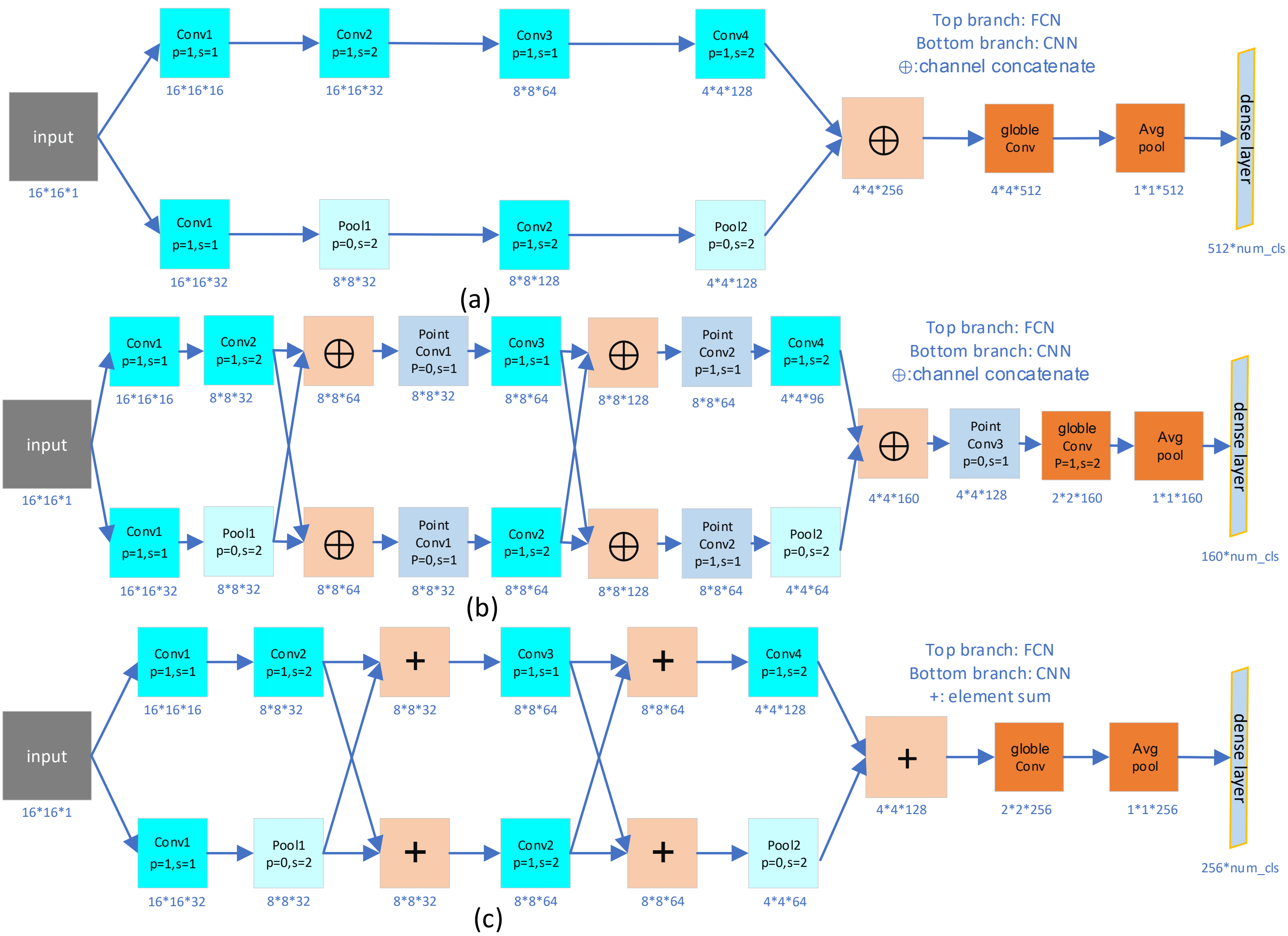


FIGURE 3. (a)Parallel Convolutional Neural Network. (b) Parallel Cross Convolutional Neural Network with Point Convolution. (c) Parallel Cross Convolutional Neural Network

the model to achieve better precision and recall for all kinds of imbalanced abnormal flow data at the same time. Table 6 shows that the proposed PCCN model is significantly better than the current hierarchical network model.

In addition, based on the experimental results in Table 3 to table 6, we can further obtain the corresponding high-order metrics to better measure our model. Specifically, the weighted average of the detection results of each category of flow in each model in table 7 to obtain 4 high-order metrics of $AA(AR)$, AP, AF and APR. The purpose of the weighted average is to make the model assign the same weight to each category of flow detection to avoid biasing the model to with larger samples, because the weighted average can more reasonably evaluate the detection performance of the model for imbalanced abnormal flows. In addition, we also calculate the geometric mean of ERci to obtain the G-mean metric to further evaluate the performance of the model. The corresponding high-order metrics detection results are shown in Table 7.

The OA in Table 7 is the overall accuracy that most researchers focus on. It reflects the overall detection performance of the model on the entire dataset. However, by comparing the average accuracy AA metric, we can find that the result of each category of abnormal flows is about 3 percentage points worse than that of the PCCN model. Through the analysis of the results in Table 7, we can find that the hierarchical network model [24] [26] [38] commonly used at present based on CNN&LSTM can indeed achieve better gain than the CNN or LSTM alone. However, the performance improvement of individual CNN and CNN&LSTM is not obvious in each metric, which means that the network model of LSTM in the hierarchical network model brings few gains. In addition, by comparing CCN and FCN networks, we can find that the detection performance of FCN is better than CNN. Therefore, the gains obtained by our proposed PCCN network model come mainly from the FCN network of the

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

TABLE 3. Detection rate of 12 categories of imbalanced abnormal flow (header)

Lable	0	1	2	3	4	5	6	7	8	9	10	11
PCCN	**1.0000**	**1.0000**	**0.7126**	0.9943	**0.9345**	**0.9910**	0.9997	**1.0000**	0.9910	**0.9990**	0.9980	**0.9896**
CCN	0.9855	**1.0000**	0.4359	0.9968	0.6730	0.9715	0.9992	**1.0000**	0.9972	0.9983	0.9975	0.9569
FCN	0.9928	**1.0000**	0.5761	0.9953	0.8756	0.9791	**1.0000**	**1.0000**	0.9972	0.9986	0.9985	0.9720
LSTM	0.9675	**1.0000**	0.3397	0.9964	0.5612	0.8212	0.9980	**1.0000**	0.9972	0.9981	0.9960	0.8881
CNN&LSTM	0.9880	**1.0000**	0.3736	**0.9983**	0.7688	0.9772	0.9990	**1.0000**	0.9944	0.9985	0.9924	0.9625
PCN	0.9928	**1.0000**	0.5444	0.9950	0.8719	0.9810	0.9995	**1.0000**	0.9963	0.9986	0.9985	0.9678
PPCCN	**1.0000**	**1.0000**	0.6892	0.9951	0.9175	0.9900	0.9997	**1.0000**	0.9981	**0.9990**	0.9995	0.9820
DPCCN	**1.0000**	**1.0000**	0.6586	0.9955	0.9183	0.9839	0.9997	**1.0000**	**0.9991**	0.9988	**0.9996**	0.9782
APCCN	0.9952	**1.0000**	0.6369	0.9924	0.9116	0.9829	0.9992	**1.0000**	0.9981	0.9988	0.9991	0.9829

TABLE 4. Precision of 12 categories of imbalanced abnormal flow (header)

Lable	0	1	2	3	4	5	6	7	8	9	10	11
PCCN	**0.9976**	**1.0000**	0.8433	**0.9871**	0.9449	**0.9812**	**0.9982**	**1.0000**	**1.0000**	**0.9998**	0.9995	0.9952
CCN	0.9927	**1.0000**	0.8428	0.9726	0.8961	0.9468	0.9928	**1.0000**	0.9944	0.9995	0.9975	0.9806
FCN	0.9928	**1.0000**	0.8285	0.9809	0.9160	0.9654	0.9955	**1.0000**	0.9972	**0.9998**	0.9987	0.9951
LSTM	0.9863	0.9999	0.8149	0.9635	0.7670	0.9068	0.9905	**1.0000**	0.9534	**0.9998**	0.9897	0.9625
CNN&LSTM	0.9903	**1.0000**	**0.8950**	0.9712	0.9272	0.9283	0.9864	**1.0000**	0.9953	0.9997	0.9993	**0.9975**
PCN	0.9904	**1.0000**	0.8221	0.9797	0.9094	0.9539	0.9953	**1.0000**	0.9972	0.9997	0.9987	0.9913
PPCCN	0.9952	**1.0000**	0.8583	0.9861	0.9461	0.9763	0.9975	**1.0000**	**1.0000**	**0.9998**	**0.9998**	0.9952
DPCCN	**0.9976**	**1.0000**	0.8566	0.9844	**0.9490**	0.9792	0.9975	**1.0000**	**1.0000**	0.9980	0.9987	0.9952
APCCN	**0.9976**	**1.0000**	0.7761	0.9839	0.9393	0.9755	0.9963	**1.0000**	0.9991	**0.9998**	0.9987	0.9947

TABLE 5. F1-Measure of 12 categories of imbalanced abnormal flow (header)

Lable	0	1	2	3	4	5	6	7	8	9	10	11
PCCN	**0.9988**	**1.0000**	**0.7825**	**0.9907**	**0.9397**	**0.9861**	**0.9990**	**1.0000**	**0.9995**	**0.9994**	**0.9996**	**0.9924**
CCN	0.9891	**1.0000**	0.5746	0.9845	0.7687	0.9590	0.9960	**1.0000**	0.9958	0.9989	0.9975	0.9686
FCN	0.9928	**1.0000**	0.6796	0.9880	0.8953	0.9722	0.9977	**1.0000**	0.9972	0.9992	0.9986	0.9834
LSTM	0.9231	**1.0000**	0.4796	0.9797	0.6171	0.8618	0.9943	**1.0000**	0.9748	0.9989	0.9929	0.9238
CNN&LSTM	0.9891	**1.0000**	0.5271	0.9846	0.8406	0.9522	0.9927	**1.0000**	0.9948	0.9991	0.9958	0.9797
PCN	0.9916	**1.0000**	0.6551	0.9873	0.8902	0.9673	0.9974	**1.0000**	0.9967	0.9992	0.9986	0.9794
PPCCN	0.9976	**1.0000**	0.7627	0.9906	0.9316	0.9821	0.9986	**1.0000**	0.9991	**0.9994**	**0.9996**	0.9885
DPCCN	**0.9988**	**1.0000**	0.7446	0.9900	0.9334	0.9715	0.9986	**1.0000**	**0.9995**	0.9983	0.9992	0.9866
APCCN	0.9964	**1.0000**	0.7011	0.9882	0.9253	0.9792	0.9977	**1.0000**	0.9986	0.9983	0.9989	0.9888

TABLE 6. EP_{c_i} of 12 categories of imbalanced abnormal flow (header)

Lable	0	1	2	3	4	5	6	7	8	9	10	11
PCCN	**0.9976**	**1.0000**	**0.6009**	**0.9815**	0.8830	**0.9724**	**0.9980**	**1.0000**	0.9910	0.9988	**0.9993**	**0.9848**
CCN	0.9784	**1.0000**	0.3674	0.9695	0.6031	0.9199	0.9920	**1.0000**	0.9916	0.9979	0.9949	0.9383
FCN	0.9856	**1.0000**	0.4773	0.9763	0.8020	0.9452	0.9955	**1.0000**	0.9944	0.9985	0.9973	0.9673
LSTM	0.8556	**1.0000**	0.2769	0.9600	0.3959	0.7446	0.9886	**1.0000**	0.9507	0.9979	0.9858	0.8548
CNN&LSTM	0.9784	**1.0000**	0.3344	0.9695	0.7128	0.9072	0.9854	**1.0000**	0.9897	0.9982	0.9917	0.9602
PCN	0.9832	**1.0000**	0.4476	0.9748	0.7929	0.9358	0.9948	**1.0000**	0.9934	0.9983	0.9973	0.9593
PPCCN	0.9952	**1.0000**	0.5884	0.9813	**0.9681**	0.9646	0.9972	**1.0000**	0.9981	**0.9989**	**0.9993**	0.9773
DPCCN	**0.9976**	**1.0000**	0.5641	0.9801	0.8714	0.9634	0.9972	**1.0000**	**0.9991**	0.9987	0.9984	0.9735
APCCN	0.9928	**1.0000**	0.4962	0.9765	0.8563	0.9589	0.9955	**1.0000**	0.9972	0.9986	0.9978	0.9777

TABLE 7. Detection results of high-order metrics (header)

Metrics	OA	$AA(AR)$	AP	AF	ARP	$G-Mean$
PCCN	**0.9917**	**0.9675**	0.9789	**0.9731**	0.9506	0.9645
CCN	0.9857	0.9177	0.9679	0.9361	0.8961	0.8955
FCN	0.9890	0.9488	0.9725	0.9587	0.9283	**0.9712**
LSTM	0.9806	0.8765	0.9529	0.8955	0.8342	0.8316
CNN&LSTM	0.9857	0.9211	0.9742	0.9380	0.9023	0.8945
PCN	0.9883	0.9455	0.9698	0.9552	0.9231	0.9342
PPCCN	0.9915	0.9642	0.9795	0.9708	**0.9557**	0.9596
DPCCN	0.9910	0.9610	**0.9797**	0.9685	0.9453	0.9553
APCCN	0.9894	0.9583	0.9718	0.9645	0.9373	0.9519

TABLE 8. Detection results of high-order metrics (payload)

Metrics	OA	$AA(AR)$	AP	AF	ARP	$G-Mean$
PCCN	**0.9987**	**0.9821**	**0.9918**	**0.9865**	**0.9742**	**0.9812**
CCN	0.9948	0.9289	0.9852	0.9609	0.9163	0.9204
FCN	0.9978	0.9676	0.9886	0.9773	0.9573	0.9655
LSTM	0.9938	0.9046	0.9797	0.9318	0.8871	0.8852
CNN&LSTM	0.9944	0.9346	0.9835	0.9561	0.9209	0.9282
PCN	0.9977	0.9675	0.9882	0.9770	0.9567	0.9656
PPCCN	0.9984	0.9756	0.9910	0.9834	0.9672	0.9741
DPCCN	0.9983	0.9758	0.9907	0.9827	0.9671	0.9745
APCCN	0.9984	0.9785	0.9906	0.9841	0.9697	0.9774

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

TABLE 9. Detection results of high-order metrics (header&payload)

Metrics	OA	$AA(AR)$	AP	AF	ARP	$G-Mean$
PCCN	**0.9992**	**0.9964**	0.9972	**0.9968**	**0.9936**	**0.9964**
CCN	0.9961	0.9771	0.9920	0.9840	0.9696	0.9731
FCN	0.9988	0.9932	0.9956	0.9944	0.9889	0.9931
LSTM	0.9959	0.9656	0.9771	0.9709	0.9442	0.9644
CNN&LSTM	0.9973	0.9851	0.9953	0.9899	0.9806	0.9845
PCN	0.9984	0.9930	0.9929	0.9930	0.9861	0.9930
PPCCN	**0.9992**	0.9958	**0.9973**	0.9966	0.9932	0.9958
DPCCN	**0.9992**	0.9960	0.9965	0.9963	0.9926	0.9960
APCCN	**0.9992**	0.9960	0.9971	0.9965	0.9930	0.9959

top branch.

Overall, our proposed PCCN model is superior to the previous network model. For both AP and APR metrics, although the PCCN model does not provide the best detection performance, it is very close to the best performance, and those models are the improved versions of PCCN. Finally, the best detection performance of $G-Mean$ is obtained in the FCN model, which also shows that the top branch can bring huge gain to the detection performance of our proposed PCCN.

3) Analysis on Features

In this section we study the detection performance of different features for imbalanced abnormal flows. We use algorithm 1 to extract the payload feature of flows and the joint feature of header and payload, called header&payload. We used the same model in the last section to evaluate the detection performance of payload and header&payload features on different models. In order to evaluate the detection results of imbalanced abnormal flow data by different features in a fair and reasonable manner, all the parameters in the experiment are set to the same as the previous section. For convenience, we only calculate the detection results of high-order metrics. Tables 4 and 5 show the detection performance of the payload and header&payload features on different models respectively.

By analyzing the experimental results in table 8, we can find that our proposed model of PCCN achieves the best results in various high-order metrics by using the payload features of imbalanced abnormal flows. Using the payload feature for detection, the gain obtained by PCCN compared with the previous models are obvious, which also proves that it is feasible to use only payload for intrusion detection and can achieve good detection results.

The experimental data in Table 9 still shows that the PCCN we proposed can obtain the best test results in all metrics. In addition, comparing the detection results of each model in Tables 4 and 5, we can find that the simultaneous extraction of header and payload features for imbalanced abnormal flow data detection can further improve the detection performance and achieve state of the art detection results. This is the same as the conclusion of the previous work [24], indicating that the simultaneous use of header and payload features can better express the information features of flows.

TABLE 10. Detection results of traditional machine learning algorithms (Header)

Metrics	OA	$AA(AR)$	AP	AF	ARP	$G-Mean$
KNN	0.9340	0.8327	0.8960	0.8571	0.7674	0.7961
NB	0.6248	0.6905	0.5903	0.5787	0.4768	0.6377
LR	0.9782	0.8746	0.9415	0.8978	0.8423	0.8310
KNN+	0.9425	0.8687	0.9352	0.8612	0.7813	0.8132
NB+	0.7326	0.6623	0.6671	0.6213	0.4854	0.6548
LR+	**0.9801**	**0.9145**	**0.9531**	**0.9124**	**0.8687**	**0.8511**
KNN++	0.9340	0.8327	0.8960	0.8571	0.7674	0.8346
NB++	0.6428	0.6905	0.5903	0.5787	0.4851	0.6375
LR++	0.9783	0.8750	0.9720	0.8982	0.8430	0.8312

In order to demonstrate the effectiveness and stability of our method, we compared the method proposed by Abdulhammed R et al. [25], who also conducted experiments on the CICIDS2017 dataset to improve the experimental results of imbalanced categories. Their method can achieve 0.997 accuracy, 0.997 f1-score and 0.988 CM_{MC} . They respectively use Auto Encoder and PCA to carry out feature dimensionality reduction, and then use traditional classifier models to classify samples. However, these reported results are not achieved on the same feature dimensionality reduction method, feature quantity and classifier, which indicates that their method lacks stability to different classification metrics. In addition, feature dimensionality reduction and classification models are not performed simultaneously so the model cannot be end-to-end trained.

4) Analysis of Oversampling and Feature Selection

We further compare PCCN with oversampling [40] [41] and feature selection algorithm [7]. First, we perform experiments with KNN, Naive Bayes (NB) and Logistic Regression (LR) algorithms. Then, we use SMOTE random oversampling algorithm to preprocess the imbalanced flow data and then use these algorithms for training, and respectively obtain three models of KNN+, NB+, and LR+. In addition, we use feature selection methods similar to [7] and obtain three feature selection algorithm models: KNN++, NB++ and LR++. Header feature is used in all the models for experiments, and the experimental results are shown in table 10.

By comparing the experimental results of table 7 and table 10, we can find that the performance of traditional machine learning algorithms is not very stable. KNN and LR are far more effective than the NB test results, but the performance is not as good as the proposed PCCN. In addition, SMOTE algorithm can be used to achieve partial performance improvement after sample balancing of the data, but the performance is still inferior to PCCN.

5) Analysis of Test Time

In order to evaluate the time performance of the model, we examine the test time of each model in the inference phase. To be fair, all our models are run on the same machine, and all the neural network models are tested with the same parameters. We experiment with the header feature, and the test time for each model is shown in table 11.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2933165, IEEE Access

TABLE 11. Test time of models (Header)

Model	Test Time(s)	Model	Test Time(s)
PCCN	7.65	KNN	34932
CNN	**4.85**	NB	8.72
FCN	5.38	LR	**4.62**
LSTM	5.37	KNN+	39285
CNN&LSTM	5.89	NB+	9.12
PCN	5.82	LR+	6.35
PPCCN	6.62	KNN++	41668
DPCCN	6.36	NB++	13.08
APCCN	6.82	LR++	7.28

From the experimental results in table 11, we can find that the proposed PCCN model in the test phase is only a little more than the commonly used hierarchical network model. In contrast to the previous algorithm, PCCN can not only achieve almost the same performance with a large number of test samples, but also save about 50% of the test time. This gain is mainly due to the fact that the original flow features we extract become more compact. In addition, compare with the traditional machine learning algorithm, the test time of each algorithm is very different. What's worse, the training time of traditional machine learning algorithms is far greater than the deep learning algorithms, resulting in SVM, random forest and other better performance algorithms are difficult to meet the training requirements of our big data computing. In table 10, although SMOTE algorithm can further improve the performance, it requires an additional 28143s (about 7.8h) of preprocessing time for oversampling of our data.

V. CONCLUSION

Since the original flow feature extraction method introduces a large number of 0 elements that are useless for learning, we propose an improved original flow feature extraction algorithm. The new original flow feature extraction algorithm greatly reduces the feature dimension, speeds up the convergence of the model, and still achieves good detection results. We propose a parallel cross convolutional neural network to perform three feature fusions to improve the detection results of highly imbalanced abnormal flow samples. The experimental results on the CICIDS2017 dataset show that the proposed model not only can detect the abnormal flow with a small number of samples but also has a good generalization ability for various features. Finally, different from most researchers, we analyze the detection results of models on each metric of each abnormal flow category and adopt high-order evaluation metrics to illustrate the effectiveness of the proposed model. We will evaluate the PCCN using other datasets in the future, such as CSE-CIC-IDS2018 and report the latest results in my GitHub website.

In the future work, we will try to use the deep learning algorithms to detect the novelty of flows and improve the detection performance. Specifically, we hope that the algorithm can detect the attack categories that do not appear in the dataset, which is very important for network security under the current big data environment.

REFERENCES

[1] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," Intelligent data analysis, vol. 6, no. 5, pp. 429–449, 2002.

[2] H. He and E. A. Garcia, "Learning from imbalanced data," IEEE Transactions on Knowledge & Data Engineering, no. 9, pp. 1263–1284, 2008.

[3] C. B. Lee, C. Roedel, and E. Silenok, "Detection and characterization of port scan attacks," Univeristy of California, Department of Computer Science and Engineering, 2003.

[4] M. H. Bhuyan, H. J. Kashyap, D. K. Bhattacharyya, and J. K. Kalita, "Detecting distributed denial of service attacks: methods, tools and future directions," The Computer Journal, vol. 57, no. 4, pp. 537–556, 2013.

[5] G. Zhao, K. Xu, L. Xu, and B. Wu, "Detecting apt malware infections based on malicious dns and traffic analysis," IEEE access, vol. 3, pp. 1132–1142, 2015.

[6] J. A. Copeland III, "Flow-based detection of network intrusions," Feb. 27 2007. US Patent 7,185,368.

[7] Y. Huang, Y. Li, and B. Qiang, "Internet traffic classification based on min-max ensemble feature selection," in 2016 International Joint Conference on Neural Networks (IJCNN), pp. 3485–3492, IEEE, 2016.

[8] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: a new over-sampling method in imbalanced data sets learning," in International conference on intelligent computing, pp. 878–887, Springer, 2005.

[9] X.-Y. Liu and Z.-H. Zhou, "The influence of class imbalance on cost-sensitive learning: An empirical study," in Sixth International Conference on Data Mining (ICDM'06), pp. 970–974, IEEE, 2006.

[10] M. Wasikowski and X.-w. Chen, "Combating the small sample class imbalance problem using feature selection," IEEE Transactions on knowledge and data engineering, vol. 22, no. 10, pp. 1388–1400, 2009.

[11] K. McCarthy, B. Zabar, and G. Weiss, "Does cost-sensitive learning beat sampling for classifying rare classes?," in Proceedings of the 1st international workshop on Utility-based data mining, pp. 69–77, ACM, 2005.

[12] N. Thai-Nghe, Z. Gantner, and L. Schmidt-Thieme, "Cost-sensitive learning methods for imbalanced data," in The 2010 International joint conference on neural networks (IJCNN), pp. 1–8, IEEE, 2010.

[13] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, and R. Togneri, "Cost-sensitive learning of deep feature representations from imbalanced data," IEEE transactions on neural networks and learning systems, vol. 29, no. 8, pp. 3573–3587, 2018.

[14] B. Anderson and D. McGrew, "Identifying encrypted malware traffic with contextual flow data," in Proceedings of the 2016 ACM workshop on artificial intelligence and security, pp. 35–46, ACM, 2016.

[15] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Network traffic classifier with convolutional and recurrent neural networks for internet of things," IEEE Access, vol. 5, pp. 18042–18050, 2017.

[16] M. R. Parsaei, S. M. Rostami, and R. Javidan, "A hybrid data mining approach for intrusion detection on imbalanced nsl-kdd dataset," International Journal of Advanced Computer Science and Applications, vol. 7, no. 6, pp. 20–25, 2016.

[17] B. Yan, G. Han, M. Sun, and S. Ye, "A novel region adaptive smote algorithm for intrusion detection on imbalanced problem," in 2017 3rd IEEE International Conference on Computer and Communications (ICCC), pp. 1281–1286, IEEE, 2017.

[18] H. Shi, H. Li, D. Zhang, C. Cheng, and X. Cao, "An efficient feature generation approach based on deep learning and feature selection techniques for traffic classification," Computer Networks, vol. 132, pp. 81–98, 2018.

[19] J. Shen, J. Xia, Y. Shan, and Z. Wei, "Classification model for imbalanced traffic data based on secondary feature extraction," IET Communications, vol. 11, no. 11, pp. 1725–1731, 2017.

[20] L. Peng, H. Zhang, Y. Chen, and B. Yang, "Imbalanced traffic identification using an imbalanced data gravitation-based classification model," Computer Communications, vol. 102, pp. 177–189, 2017.

[21] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in 2015 military communications and information systems conference (MilCIS), pp. 1–6, IEEE, 2015.

[22] Z. Liu, R. Wang, M. Tao, and X. Cai, "A class-oriented feature selection approach for multi-class imbalanced network traffic datasets based on local and global metrics fusion," Neurocomputing, vol. 168, pp. 365–381, 2015.

[23] Z. Cui, F. Xue, X. Cai, Y. Cao, G.-g. Wang, and J. Chen, "Detection of malicious code variants based on deep learning," IEEE Transactions on Industrial Informatics, vol. 14, no. 7, pp. 3187–3196, 2018.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.

[24] Y. Zhang, X. Chen, L. Jin, X. Wang, and D. Guo, "Network intrusion detection: Based on deep hierarchical network and original flow data," IEEE Access, vol. 7, pp. 37004–37016, 2019.

[25] R. Abdulhammed, H. Musafer, A. Alessa, M. Faezipour, and A. Abuzneid, "Features dimensionality reduction approaches for machine learning based network intrusion detection," Electronics, vol. 8, no. 3, p. 322, 2019.

[26] W. Wang, Y. Sheng, J. Wang, X. Zeng, X. Ye, Y. Huang, and M. Zhu, "Hast-ids: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection," IEEE Access, vol. 6, pp. 1792–1806, 2018.

[27] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3431–3440, 2015.

[28] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117–2125, 2017.

[29] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," arXiv preprint arXiv:1502.03167, 2015.

[30] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization.," in ICISSP, pp. 108–116, 2018.

[31] S. M. H. Bamakan, H. Wang, and Y. Shi, "Ramp loss k-support vector classification-regression; a robust and sparse multi-class approach to the intrusion detection problem," Knowledge-Based Systems, vol. 126, pp. 113–126, 2017.

[32] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," Information Processing & Management, vol. 45, no. 4, pp. 427–437, 2009.

[33] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," 2011.

[34] M. Yeo, Y. Koo, Y. Yoon, T. Hwang, J. Ryu, J. Song, and C. Park, "Flow-based malware detection using convolutional neural network," in 2018 International Conference on Information Networking (ICOIN), pp. 910–913, IEEE, 2018.

[35] W. Wang, M. Zhu, J. Wang, X. Zeng, and Z. Yang, "End-to-end encrypted traffic classification with one-dimensional convolution neural networks," in 2017 IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 43–48, IEEE, 2017.

[36] A. Azzouni and G. Pujolle, "A long short-term memory recurrent neural network framework for network traffic matrix prediction," arXiv preprint arXiv:1705.05690, 2017.

[37] X. Yuan, C. Li, and X. Li, "Deepdefense: identifying ddos attack via deep learning," in 2017 IEEE International Conference on Smart Computing (SMARTCOMP), pp. 1–8, IEEE, 2017.

[38] 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.

[39] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," IEEE transactions on pattern analysis and machine intelligence, vol. 40, no. 4, pp. 834–848, 2018.

[40] S. Hu, Y. Liang, L. Ma, and Y. He, "Msmote: Improving classification performance when training data is imbalanced," in 2009 second international workshop on computer science and engineering, vol. 2, pp. 13–17, IEEE, 2009.

[41] A. Tesfahun and D. L. Bhaskari, "Intrusion detection using random forests classifier with smote and feature reduction," in 2013 International Conference on Cloud & Ubiquitous Computing & Emerging Technologies, pp. 127–132, IEEE, 2013.

YONG ZHANG has been an associate professor in the School of Electronic Engineering at Beijing University of Posts and Telecommunications, P.R.C. He holds a Ph.D. degree from Beijing University of Posts and Telecommunications. His research interests include self-organizing networks, mobile communications, and cognitive networks.

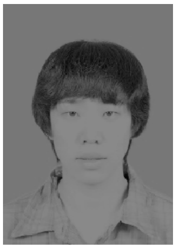
XU CHEN received the B.S. degree in communication engineering from Chongqing University of Posts and Telecommunications, Chongqing, China, in 2017. He is currently pursuing the B.S. degree with the Beijing University of Posts and Telecommunications, Beijing, China. His research interests include deep learning and intrusion detection.

DA GUO received his Ph.D. degree in electrical engineering from Beijing University of Posts and Telecommunications, and he is currently a senior engineer at that institution. His research interests are in mobile communications, opportunistic networks, WSN, and P2P networks.

MEI SONG is a Professor in Beijing University of Posts and Telecommunications. She holds a Ph.D degree in Beijing University of Posts and Telecommunications. Her current research interests include resource allocation and mobility management in heterogeneous and cognitive network, cooperative communication, and other advanced technology in future communication.

YINGLEI TENG received the B.S. degree from Shandong University, China, in 2005, and the Ph.D. degree in electrical engineering from the Beijing University of Posts and Telecommunications (BUPT) in 2011. She worked as the post-doctor in HKUST, Hong Kong, in 2015. She is currently an Associate Professor with the School of Electronic Engineering, BUPT. Her current research interests include UDNs and massive MIMO, IoTs and blockchains.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.



XIAOJUAN WANG received the Ph.D degree in electrical and electronics engineering from Beijing University of Posts and Telecommunications, Beijing, China. She is currently the associate professor in the school of electronic engineering from Beijing University of Posts and Telecommunications. Her current research interest is artificial intelligence and so on.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.