Computer Science and Information Systems 19(1):309–326 https://doi.org/10.2298/CSIS210617055X

RICNN: A ResNet&Inception Convolutional Neural Network for Intrusion Detection of Abnormal Traffic

Benhui Xia[1], Dezhi Han[1], Ximing Yin[2], and Na Gao[1]

[1] College of Information Engineering, Shanghai Maritime University
200031 Shanghai, China
{201930310092}@stu.shmtu.edu.cn, dzhan@shmtu.edu.cn
[2] The Third Research Institute of Ministry of Public Security
201306 Shanghai, China

Abstract. To secure cloud computing and outsourced data while meeting the requirements of automation, many intrusion detection schemes based on deep learning are proposed. Though the detection rate of many network intrusion detection solutions can be quite high nowadays, their identification accuracy on imbalanced abnormal network traffic still remains low. Therefore, this paper proposes a ResNet &Inception-based convolutional neural network (RICNN) model to abnormal traffic classification. RICNN can learn more traffic features through the Inception unit, and the degradation problem of the network is eliminated through the direct mapping unit of ResNet, thus the improvement of the model's generalization ability can be achievable. In addition, to simplify the network, an improved version of RICNN, which makes it possible to reduce the number of parameters that need to be learnt without degrading identification accuracy, is also proposed in this paper. The experimental results on the dataset CICIDS2017 show that RICNN not only achieves an overall accuracy of 99.386% but also has a high detection rate across different categories, especially for small samples. The comparison experiments show that the recognition rate of RICNN outperforms a variety of CNN models and RNN models, and the best detection accuracy can be achieved.

Keywords: Intrusion Detection, ResNet, Inception, CNN, Traffic Classification, Imbalanced Samples.

1. Introduction

With the maturity of cloud computing technology, more and more outsourced data are stored in the cloud [1–4]. Worse still, the wrongdoers are tempted by the enormous value of digital asserts such as users' privacy and transaction records in this era of cloud computing [5]. They thus constantly attack the network for economic benefits [6]. To ensure the security of cloud space, intrusion detection technology is needed to secure outsourced data traffic [7]. As a defense means, network traffic detection technology can identify the abnormal data in the byte stream, so that the system managers can find the attack behaviors in time, and then take corresponding measures to resist them and therefore reduce the loss. While early intrusion detection techniques relied on manual extraction of traffic features, not only the chosen algorithm but the pre-defined set of features could make a significant influence on its recognition accuracy [8, 9]. So far, with the development of deep learning, especially the increasing maturity of convolutional neural networks (CNN) on image recognition, many researchers have introduced neural networks into the field of network traffic detection, and have made numerous achievements. However, following problems can be generally found in existing researches. First, the datasets used by many researchers are too old [10]. In recent years, new types of attack have been emerging, but many public datasets are not adequate for the current cyberspace, either in terms of variety or quantity. Second, many datasets are processed data that has lost the full information of the original byte flow [11], resulting in solution's failure to simulate real detection environment. Third, many researchers have ignored the imbalanced distribution of different kinds of abnormal categories in the dataset [12]. As a result, models with high overall accuracy can be quite inaccurate when it comes to small samples.

Notably, data imbalance is an important factor for machine learning [13]. In the field of intrusion detection, different anomalous flows vary greatly in terms of structure, behavior, etc. For example, attacks such as port scanning [14] or DoS [15] are easy to be detected and captured, so such anomalous flows often take up a large portion of the byte stream. In contrast, some complex attack types, such as APT [16], that are difficult to be collected only account for a small proportion in the dataset. To solve the impact of imbalanced data distribution on recognition accuracy, this paper chooses to use the original byte traffic as input directly. The flow form composed of the same five-tuple features [17] (i.e., source IP address, destination IP address, source port, destination port, and protocol) maximally preserves the structure and spatial features of each abnormal flow itself. These features are then expressed by a traffic grayscale map, thus, with the help of a CNN-based neural network, different abnormal categories can be identified through the automatic extraction of them. Finally, we choose CICIDS2017 [10], which provides raw byte stream files and contains many different types of attack flow, as the dataset for our experiments. Statistically, the percentage of different malicious flow types varies greatly, also, the attack categories match the realistic cyberspace environment and thus fit the scope of our study.

Many studies at this stage often use a single-path CNN network structure [18], which often fails to extract enough features from the grayscale map. Besides, simple increasement of the network depth may lead to the downgrade of model accuracy. In order to improve the generalization ability of the model and to deal with gradient disappearance, this paper proposes a ResNet&Inception convolutional neural network (RICNN) by combining the advantages of residual networks (ResNet) [19] and Inception feature fusion [20]. RICNN adopts parallel structures for feature extraction of the input to obtain more feature maps in the form of feature fusion. And direct mapping via ResNet [19] will be used to solve the problem of gradient disappearance during learning process. The final experimental results show that RICNN achieves a recognition accuracy of over 99% on the dataset CICIDS2017 and has high recognition rate on small samples categories.

The main contributions of this paper are as follows:

(1) A new network model, RICNN, is proposed, which can effectively improve the recognition precision of small samples in multi-classification detection of abnormal traffic.

(2) An improvement model, ICNN, is proposed, which achieves the purpose of simplifying the network structure and does not affect the detection accuracy.

(3) The experimental results on the dataset CICIDS2017 show that RICNN not only has the overall accuracy of 99.386%, but also has high detection precision in different

categories, especially those with small samples. The comparison experiments show that the recognition rate of RICNN outperforms a variety of CNN models and RNN models, which can achieve the highest detection accuracy.

The rest of this paper is organized as follows. Section 2 discusses related works in this field. Section 3 describes the data pre-processing steps and the specific structure of the proposed model in this paper. Section 4 conducts an experimental comparison of our proposed model on CICIDS2017 to assess the effectiveness of our model. Finally, we draw a conclusion in Section 5.

2. Related Work

Current supervised learning models for classifying abnormal traffic are mainly divided into using CNN models to extract spatial features and using RNN models to extract timing features [18]. Most studies, though can obtain high detection accuracy, fail to take into account the impact of the imbalanced distributed dataset on the classification results.

Marín, G et al. [21] investigated the accuracy of deep learning models for recognition at the packet level and the flow level. The results showed that the recognition rate was higher for inputs in flow form. However, the authors did not consider the impact of small sample factors on the model, and also the experiments had a small variety of malicious samples. And many researchers raised the detection accuracy by improving the CNN model. Jing Ran et al. [22] first used 3-dimensional CNN networks for network traffic classification. The authors applied video analysis to traffic recognition by processing vector time series and one-hot coding to form a fixed-length 3-dimensional input, which was learned by a 3-dimensional CNN structure with features in time and space. The result showed that this model had a higher detection accuracy on multiple categories. Hyun-Kyo Lim et al. [23] introduced the residual structure to the CNN network. After experimental comparison, it was found that the ResNet model could improve the generalization ability of the network. When the input data contained more information, the ResNet model had higher accuracy. Wei Wang et al. [24] proposed to use a 1-dimensional CNN model to classify encrypted traffic, and experiments proved that 1D-CNN performed better in end-to-end encrypted traffic recognition. Yong Zhang et al. [25] proposed a parallel CNN structure, PCCN, which improved the generalization ability of the network through feature fusion without increasing the depth of the network. Peng Yujie et al. [26] proposed a 1.5D-CNN model that combined 2D-CNN and 1D-CNN to extract features of traffic in different dimensions, and its multi-classification accuracy reached 98.5%. Samson Ho et al. [27] used a modified version of LeNet which did not take into account the imbalanced dataset, so the model did not have a high detection rate for small samples at multiple classifications. In the same way, researchers have focused on the impact of hybrid models on anomaly detection. Manuel Lopez-Martin et al. [28] studied the influence of timing features in the network stream on the classification detection results. The use of CNN and LSTM for vector time series data reduced the impact of feature engineering, but experiments showed that the model's recognition accuracy was still low for certain packet types. Monika Roopak et al. [29] proposed a model that combined RNN and CNN to detect DDoS attacks on IoT networks. Although its detection accuracy reached 97.16%, the author only considered DDoS attack, and did not verify the detection accuracy of the model under multiple attacks. Jiayin Feng et al. [30] proposed a model combining cascades CNN and autoencoder for mobile terminal intrusion detection to classify mobile traffic in a semi-supervised form. But the model performed poorly in the case of multiple classifications. Khan, M. A. et al. [31] proposed a hybrid model of 1D-CNN and two-layer LSTM and achieved 97.29% accuracy on the dataset ISCX2012 [32]. Pengfei Sun et al. [33] also used a hybrid model of CNN and LSTM, and eliminated the effect of sample category imbalance on the model by weight optimization. The accuracy of the model in multiple classifications reached 98.67%. Kaiyuan Jiang et al. [12] used a hybrid sampling method to solve the dataset imbalance problem by reducing the noise in large samples through one-sided selection (OSS), and increasing small samples by using the synthetic minority oversampling technique (SMOTE) to finally build a relatively balanced dataset. However, the authors' hybrid model using CNN-BiLSTM failed to achieve the required recognition accuracy on the dataset. Maonan Wang et al. [34] combined CNN and stacked autoencoder (SAE), with CNN automatically extracting high-level features from the original traffic, SAE encoding 26 statistical features, and finally merging them into new high-level features. The framework achieved 98% accuracy in the multi-classification of encrypted traffic, but it took a lot of time to perform feature statistics on the original flow, which required high labor costs. Wanqian Zhang et al. [35] explored the relationship between window size and model classification accuracy. By detecting CPU occupancy in real time and finding appropriate dynamic parameters, the recognition time of CNN model was reduced, and a new framework of online traffic detection technology was realized. Chongzhen Zhang et al. [36] proposed a general intrusion detection framework that used an unsupervised autoencoder for feature extraction, and the extracted low-dimensional recombined features were stored for testing and retraining. However, the results showed that the recognition rate of small samples was lower than other sample categories in multi-classification.

Compared with the above schemes, our model has the following advantages. Our model has higher accuracy. And under the condition of imbalance distribution of data samples, the detection effect of all malicious categories is optimal. Our model structure does not deepen the number of network layers, but uses multiple branches to process the features. In terms of data processing, we combine the pre-processing methods of [11] and [25] to segment the raw traffic into flow format and form fixed-length input samples. Two types of training samples are generated, one is to extract the header of the packet and the other is to extract the payload of the packet, and they share the same length. Instead of deepening the layers of the network to improve the detection accuracy on small sample categories, we improve the parallel units of [25] and introduce ResNet to enhance the generalization ability of the model.

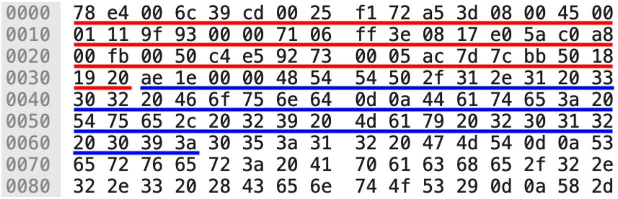
3. Model and Method

In this section, we design a hybrid neural network model to implement the detection of abnormal traffic. The proposed model primarily combines two convolutional neural networks, ResNet and Inception, and uses feature fusion to achieve accurate recognition of small samples. Our model uses raw traffic as input and automatically extracts the original features from abnormal traffic to complete the multi-classification. For the model to better learn the spatial features of the original traffic, we transform the samples into twodimensional grayscale maps. First, we will introduce the pre-processing process for the dataset.

3.1. Data Preprocessing

In this paper, the original traffic file is processed directly to obtain sample data in the flow format with a fixed length of 256 bytes. The process is as follows.

(1) **Raw file processing.** The dataset is a large PCAP file. We use the SplitCap tool [37] to merge packets with the same five tuples into a single flow. We limit the number of packets in a flow to five, and when the number is over five, we form a new flow where vacancies are filled with 0 bytes.



```
0000  78 e4 00 6c 39 cd 00 25  f1 72 a5 3d 08 00 45 00
0010  01 11 9f 93 00 00 71 06  ff 3e 08 17 e0 5a c0 a8
0020  00 fb 00 50 c4 e5 92 73  00 05 ac 7d 7c bb 50 18
0030  19 20 ae 1e 00 00 48 54  54 50 2f 31 2e 31 20 33
0040  30 32 20 46 6f 75 6e 64  0d 0a 44 61 74 65 3a 20
0050  54 75 65 2c 20 32 39 20  4d 61 79 20 32 30 31 32
0060  20 30 39 3a 30 35 3a 31  32 20 47 4d 54 0d 0a 53
0070  65 72 76 65 72 3a 20 41  70 61 63 68 65 2f 32 2e
0080  32 2e 33 20 28 43 65 6e  74 4f 53 29 0d 0a 58 2d
```

Fig. 1. Packet interception. The first 50 bytes are header features (red underline), and the 51st-100th bytes are payload features (blue underline)

(2) **Packet processing.** As shown in Fig. 1, to make the final training samples in the same length, the packets need to be intercepted and padded. The first 50 bytes of the packet are used as the header features and the 51st to 100th bytes are used as the payload features, with any shortfall in length being padded with “0”. Statistical analysis shows that a length of 50 bytes can contain most of the traffic features [11]. The number “256” is used to divide each packet in a flow sample, as it equals to 0 when being transformed into a grayscale map, it will not affect the original features of the flow.

(3) **Grayscale map conversion.** Since the same type of network attack flow has a similar structure [38], and the original bytes of the packets take values within 0-255, the data can therefore be converted into a grayscale map. In this way, the classification of different malicious flows can be realized by transforming the process of extracting traffic structure features into that of extracting texture features from grayscale maps. In this paper, the flow samples with 256 bytes in length are converted to 16*16 grayscale maps.

3.2. Model Design

Because the various types of samples in the dataset are imbalanced distribution, we choose CNN to extract features from the samples to improve the recognition accuracy for small samples. The CNN-based models have good recognition capability in image classification [39]. In this paper, we propose a RICNN model combining ResNet and Inception to improve the detection rate of abnormal traffic with imbalanced data. As shown in Fig. 2, the proposed model is mainly composed of three residual blocks, each of which is an Inception structure that learns more spatial features from the samples through feature fusion to achieve the detection of traffic categories.

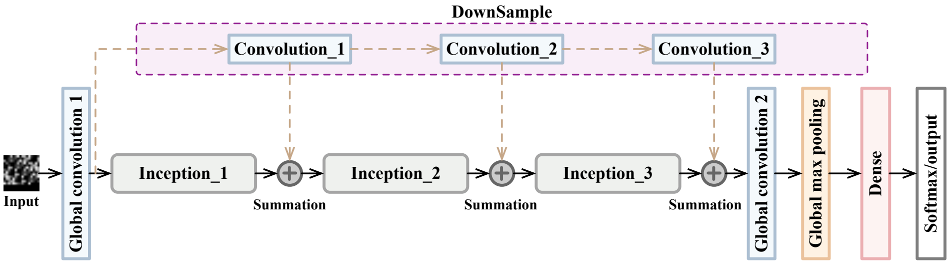


Fig. 2. RICNN network model architecture. Its input is a 16 × 16 grayscale matrix. The main structure of RICNN is three residual blocks, each of which consists of an Inception unit and a direct mapping. In addition, there are several global network layers to increase and reduce the dimension of the feature maps, and finally the model prediction results are output through the Softmax layer

(1) Inception Unit

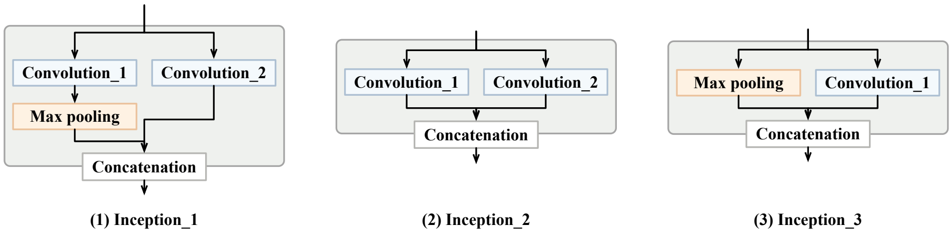


Fig. 3. Details of three Inception units

Inception is a CNN functional unit proposed by Google [20], which sets up different branching structures in the same block. Each branch extracts features from the original maps in parallel and extracting more features with different receptive field sizes without increasing the depth of the network. For a feature map X, its dimension is $H \times W \times C$. Assuming that a Inception unit has n branches, the height, width, and channel number of the output feature maps of i-th branch are h, w and c_i, then the final concatenation operation of each Inception is as follows:

$$H_{out} \times W_{out} \times C_{out} = h \times w \times \sum_{i=1}^{n} c_i \tag{1}$$

Table 1. Parameters of the Inception units

Name	Branch	Operation	Input Size	Convolution Kernel	Step	Padding	Output Size
Inception_1	Branch_1	Conv	16*16*16	3*3	2	1	8*8*32
	Branch_2	Conv	16*16*16	3*3	1	1	16*16*32
		MaxPool	16*16*32	2*2	2	0	8*8*32
Inception_2	Branch_1	Conv	8*8*64	3*3	1	1	8*8*96
	Branch_2	Conv	8*8*64	3*3	1	1	8*8*96
Inception_3	Branch_1	Conv	8*8*192	3*3	2	1	4*4*256
	Branch_2	MaxPool	8*8*192	2*2	2	0	4*4*192

Tab. 1 shows the structural parameters of the three Inception units. As shown in Fig. 3, the branches of Inception consist of convolutional layers and maximum pooling layers. The first Inception unit has one convolutional layer in each of the two branches. In the first branch, the convolutional layer is responsible for sampling and dimension reduction, but the convolutional layer in the second branch is just responsible for sampling, and the maximum pooling layer performs the dimension reduction and preserves the texture features of the grayscale maps. The second Inception unit consists of two convolutional layers, which form a parallel branch structure. The first branch in the third Inception unit is a convolutional layer, which is responsible for sampling and dimension reduction, and the second branch is a maximum pooling layer for texture reduction and preservation.

(2) ResNet Unit

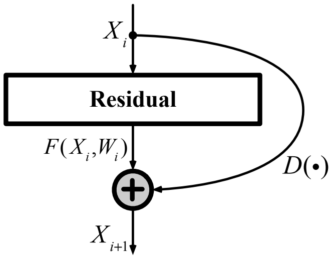


Fig. 4. Residual block structure

The residual network is designed to solve the problem of vanishing gradient and accuracy degradation in deep networks so that each layer of the network can fully learn the original traffic features and improve the classification accuracy of malicious traffic. As shown in Fig. 4, a residual block mainly consists of a residual part and a direct mapping. For an input feature map X_i, its residual part is $F(x_i, W_i)$, and W_i denotes the set of convolution operations. The direct mapping is:

$$D(x) = w' * x \tag{2}$$

w' represents a 1*1 convolution operation, which reduces the dimension of the input feature map to be consistent with the output dimension of the residual part. The final output of the residual block is:

$$X_{i+1} = \lambda_{relu}(D(X_i) + F(X_i, W_i)) \tag{3}$$

λ_{relu} represents the activation function ReLU, which can improve the nonlinear ability of the network. Through direct mapping $D(X_i)$, we can solve the problem of network learning difficulties. The residual part $F(X_i, W_i)$ of the proposed model is composed of Inception and three direct mapping branches. Each branch uses a 1*1 convolution layer to increase and reduce the dimension of the original feature maps, so that the output is consistent with the residual part.

(3) Overall Model Architecture

The proposed model consists mainly of convolutional layers and maximum pooling layers. The convolution layers are used for feature sampling on the input maps and expanding the dimension of the output maps. A convolution kernel ω with the size of $f * f$ is sampled on the input map X_i, and the output feature map is:

$$X_{i+1} = \lambda_{relu}(BN(\omega * X_i)) \tag{4}$$

For each batch, all feature maps need to be batch normalization (BN) before nonlinear activation. First, we normalize the feature data with each mini-batch as a unit:

$$\begin{aligned} X_i &= \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} \\ &= \frac{x_i - \frac{1}{m}\sum\limits_{i=1}^{m} x_i}{\sqrt{\frac{1}{m}\sum\limits_{i=1}^{m}\left(x_i - \frac{1}{m}\sum\limits_{i=1}^{m} x_i\right)^2 + \varepsilon}} \end{aligned} \tag{5}$$

Then move and scale the feature, that is:

$$BN_{\gamma,\beta}(X) = \gamma X + \beta \tag{6}$$

Batch normalization improves generalization ability of the network and increases the accuracy of learning, which also has the effect of preventing overfitting [40].

Maximum pooling is used for the pooling layer. The recognition of the traffic grayscale map mainly relies on the learned texture features, that the texture composed of light-colored pixel units, while weakening the dark part of the grayscale map. Therefore, when down-sampling the feature map, maximum pooling can preserve the texture features very well and reduce the loss during the learning process.

Tab. 2 shows the other parameters of the proposed model. The original traffic map is fed into the first global convolutional layer to obtain initial sampled feature maps with an expanded number of channels. The feature maps are then fed into the residual block structure, where the different features are extracted by the Inception unit in parallel, and the accuracy degradation problem is eliminated by the direct mapping. The second global convolution of the model is the expansion of the feature maps, and the global pooling layer reduces the dimension of the feature maps and the number of parameters. The final

Table 2. Parameters of the overall model

Operation	Input Size	Convolution kernel	Step	Padding	Output Size
Global Convolution 1	16*16*1	3*3	1	1	16*16*16
Global Convolution 2	4*4*448	3*3	1	1	4*4*896
Global Max Pooling	4*4*896	4*4	1	0	1*1*896
Dense	896	-	-	-	12

fully-connected and softmax layer map the final extracted high-dimensional features into specific categories, enabling multiple classification of malicious traffic.

3.3. ICNN

To simplify the network structure, we propose ICNN, an improved version based on the model in this paper. As shown in Fig. 5, in the improved version, we remove the residual block and add a 1*1 convolution layer to each Inception unit instead of direct mapping, and perform features fusion with other branches to achieve improved network generalization. Simultaneously, we abandon a global convolution network layer, which reduces the number of learning parameters.

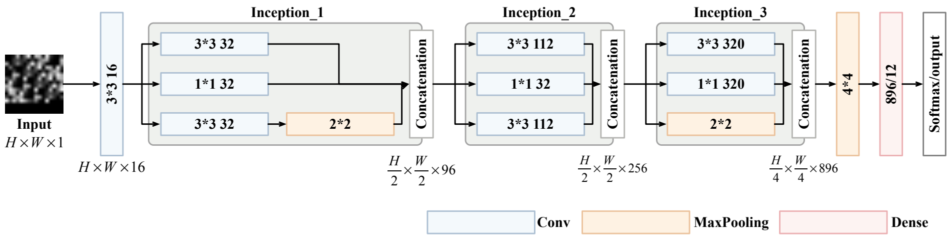


Fig. 5. Improved model based on Inception (ICNN)

4. Experimental Evaluation

In this section, we perform an experimental validation of the proposed model and test the validity of our model through contrast experiments. We choose CICIDS2017 as the dataset for our experiments. Our experimental environment is shown in Tab. 3.

4.1. Dataset

The dataset chosen for this paper needs to have the following characteristics: (I) It is the original traffic; (II) It contains various types of attack; (III) The distribution of all kinds of traffic is imbalanced; (IV) It is a relatively new dataset. Nowadays, many datasets are too

Table 3. Experimental environment parameters.

Name	Parameters
CPU	Intel(R) Xeon(R) Silver 4116 CPU @ 2.10GHz
GPU	NVIDIA Quadro P4000
RAM	64GB
OS	Ubuntu 16.04

old and lack of new attack types. Similarly, some datasets only contain partial features of packets, and lack of complete traffic data. After comprehensive consideration, we choose CICIDS2017 [10] as the experimental dataset for this paper.

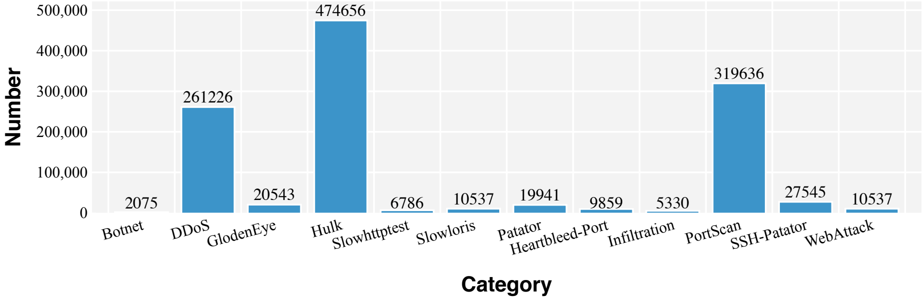


Fig. 6. Statistics on the number of various attack types in the CICIDS2017 dataset

CICIDS2017 is an open source network intrusion detection dataset, which is composed of traffic data collected by Canadian Network Security Agency over five consecutive days in 2017, and has completely marked various types of traffic. According to the statistical analysis, the dataset contains a total of 12 different types of attack traffic, and the number of different attacks is imbalanced. As shown in Fig. 6, Hulk, DDoS, and PortScan account for 90% of the overall dataset, while types such as Botnet and Infiltration make up much small percentage of the attacks. We divide the CICIDS2017 dataset, 80% of which are used as training set and 20% as testing set. To make each category equally distributed in the training set and testing set, we need to divide each type with a 4:1 ratio.

4.2. Evaluation Indicators

To analyze the detection accuracy of the model for each category of abnormal traffic, we choose Accuracy Rate (Acc) as the indicator of the overall model and Precision, Recall and F1 values as the recognition indicators for each category. For the category i of abnormal traffic, the following four indicators can be calculated:

True Positives (TP_i): The predicted category is i, and the true category is i as well.

False Positives (FP_i): The predicted category is i, but the true category is not i.

False Negatives (FN_i): The predicted category is not i, but the true one is i.

Then, for the model with n-classification, the calculation formula of Acc, which represents the overall recognition accuracy of the model, is as follows:

$$Acc = \frac{\sum\limits_{i=1}^{n} TP_i}{\sum\limits_{i=1}^{n} (TP_i + FN_i)} \times 100\% \tag{7}$$

For the evaluation indicators Precision and Recall for category i, the formulae are:

$$P_i = \frac{TP_i}{TP_i + FP_i} \times 100\% \tag{8}$$

$$R_i = \frac{TP_i}{TP_i + FN_i} \times 100\% \tag{9}$$

The F1 indicator for category i is calculated as:

$$F1_i = \frac{2 \times P_i \times R_i}{P_i + R_i} \times 100\% \tag{10}$$

Accuracy (Acc) is the ratio of the number of correctly identified samples to the overall samples. It measures the general classification effect of the model, but it is not specific to the recognition accuracy of a certain category. For this paper, it is important to focus not only on the general classification effect, but also on the identification of specific categories. In contrast, Precision, Recall and F1 focus more attention on evaluating the detection effectiveness of the model on different categories. The use of the above indicators provide a comprehensive and realistic assessment of our model.

4.3. Result Analysis

To investigate the detection capability of the proposed model on the imbalanced abnormal traffic dataset, we compare the performance indicators of it with other CNN models. We conduct experiments on the header and payload of samples to investigate the recognition ability of the proposed model on different segments of raw traffic features. And to investigate the effectiveness of our model in extracting features directly on the original traffic, we also compare it with LSTM and CNN+LSTM models that identify the timing properties of the samples. Furthermore, in order to simplify the proposed network, we propose an improved network ICNN that uses only the Inception structure to achieve extremely high detection rates and improve the operational efficiency of the model.

(1) Experimental content

Table 4. Variation of learning rate with epoch

Epoch	1-5	6-8	9-10
Learning Rate	0.0001	0.00001	0.000001

In the training phase, we set the epoch to 10 and the mini-batch for each round is 256. Tab. 4 shows the setting of the learning rate for different epoches. At the same time, after each epoch of training, we use the testing set to test the model and obtain the actual accuracy in the process of model learning.

(2) Analysis of experimental results

We use the proposed RICNN model in this paper as the basis for performance comparisons with other network models, and also compare the experimental performance of ICNN. For other CNN models, we choose 1d-CNN [24], 2d-CNN [27] and PCCN [25]. Meanwhile, we choose RNN models like LSTM [41,42] and CNN+LSTM [28,29,33] to compare their effectiveness in extracting timing features from the original traffic samples.

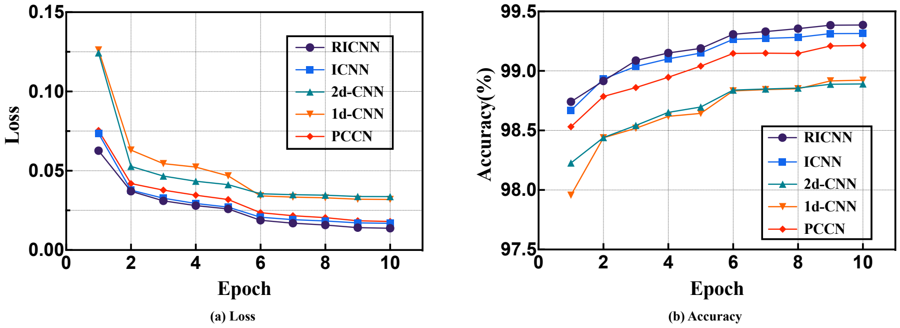


Fig. 7. Comparison of five CNN models (header). (a) indicates the variation of loss with epoch, and (b) indicates the variation of accuracy with epoch

To show the ability of RICNN in extracting the texture features from maps and obtaining the spatial features of the original traffic, three different CNN models are used to compare the detection accuracy of the 12-classification of abnormal traffic samples (header) with RICNN and ICNN. Fig. 7 (a) shows the variation of loss value with epoch in the process of model learning, and Fig. 7 (b) shows the change of accuracy. It can be seen that RICNN, ICNN, and PCCN which includes parallel feature extraction branches are more than 0.3% higher than the 1d-CNN and 2d-CNN with a single extraction path in terms of abnormal traffic detection accuracy. Due to the improved generalization of the network by direct mapping, RICNN and ICNN had a 0.17% higher detection accuracy than PCCN.

Raw traffic data holds the most complete flow features, and feature learning directly in the raw traffic can improve the accuracy of classification. LSTM can extract the timing features of the samples to distinguish between different malicious traffic. However, the comparison of the indicators in Fig. 8 shows that in BotNet, Goldeneye, Slowhttp, Infiltration and Web Attack, the number of samples is so small that the recognition accuracy in these categories is 0, that is, LSTM does not learn the correct features at all. The CNN+LSTM hybrid model has improved the recognition accuracy of the above abnormal traffic types (except BotNet) because of the spatial features extracted by CNN.

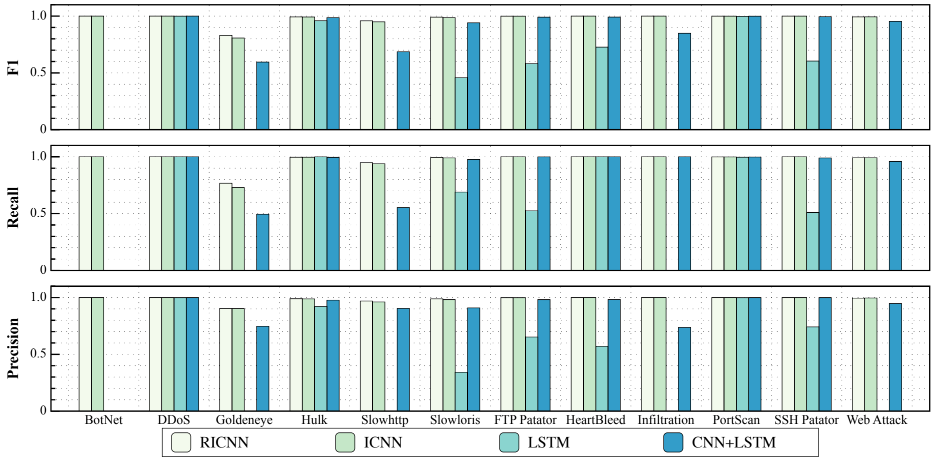


Fig. 8. Comparison of evaluation indicators between the proposed models and the RNN models for imbalanced abnormal traffic (header)

The above comparative experiments show that the original traffic needs to be specifically encoded [43, 44], and the RNN models can extract complete timing features. However, encoding also irreversibly corrupts the original traffic similar to hand-extracted features and is therefore less effective in anomaly detection for small samples.

Table 5. Precision of 12 categories of imbalanced traffic (payload). Numbers 1-12 respectively indicate: Botnet, DDoS, GlodenEye, Hulk, Slowhttptest, Slowloris, Patator, Heartbleed-Port, Infiltration, PortScan, SSH-Patator, WebAttack

Model	1	2	3	4	5	6	7	8	9	10	11	12
RICNN	0.9787	0.9963	0.9976	1.0000	0.9978	0.9915	1.0000	0.9980	0.9467	1.0000	0.9993	0.9995
ICNN	0.9921	0.9959	0.9973	0.9999	0.9985	0.9920	1.0000	0.9990	0.9605	1.0000	0.9996	0.9995
PCCN	0.9814	0.9965	0.9968	0.9999	0.9985	0.9925	1.0000	0.9995	0.9542	1.0000	0.9993	0.9995

Finally, we investigate the effect of the payload part on the detection accuracy of the model when it is used as training data. The header part of the packet mainly contains header information, including protocol, address, etc., and its structure is relatively fixed. Furthermore, the payload part contains the application layer data of the packet, which represents the real information and better expresses the feature information of the abnormal traffic. We choose PCCN, which also has parallel structures, for comparison with the two proposed models. Tab. 5-7 show the Precision, Recall and F1 score of the three models in different abnormal classes. The tables show that even the BotNet and Infiltration categories, which have the smallest number of samples, have improved recognition rates, indicating that RICNN and ICNN can have higher recognition rates on the pay-

Table 6. Recall of 12 categories of imbalanced traffic (payload). The category number is consistent with Tab 5

Model	1	2	3	4	5	6	7	8	9	10	11	12
RICNN	0.8867	0.9997	0.9998	0.9985	0.9816	0.9991	0.9992	0.9995	0.9493	0.9999	0.9989	0.9995
ICNN	0.9108	0.9998	0.9976	0.9985	0.9838	0.9995	0.9992	0.9995	0.9343	0.9999	0.9995	0.9991
PCCN	0.8916	0.9998	0.9983	0.9985	0.9816	0.9991	0.9992	0.9990	0.9568	0.9999	0.9998	0.9995

Table 7. F1-score of 12 categories of imbalanced traffic (payload). The category number is consistent with Tab 5

Model	1	2	3	4	5	6	7	8	9	10	11	12
RICNN	0.9305	0.9980	0.9987	0.9993	0.9896	0.9953	0.9996	0.9987	0.9480	1.0000	0.9991	0.9995
ICNN	0.9497	0.9979	0.9974	0.9992	0.9911	0.9957	0.9996	0.9992	0.9472	0.9999	0.9995	0.9993
PCCN	0.9343	0.9981	0.9976	0.9992	0.9900	0.9957	0.9996	0.9992	0.9555	1.0000	0.9995	0.9995

load dataset. Fig. 9 shows the variation of the three models with epoch. It can be seen that ICNN learns features quickly and achieves higher accuracy, while RICNN is less accurate than ICNN and PCCN in the first few epoches due to the more complex network structure. Finally, at the 10th epoch, all three models achieved a detection accuracy of 99.87%. With limited resources, ICNN has more advantages.

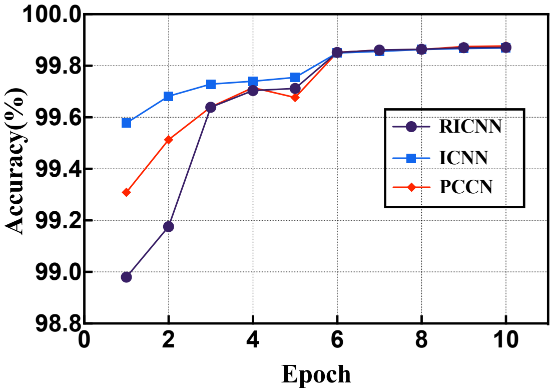


Fig. 9. Accuracy comparison of three parallel CNN multi-classification models (payload)

5. Conclusion

We want to retain the maximal amount of raw traffic features, so the raw packets are cut into header and payload parts of a specific length and then combined into separate flow forms as input data. This paper proposes a convolutional neural network based on ResNet and Inception. Through three times of feature fusion and direct mapping, the detection accuracy of imbalanced abnormal samples is improved without increasing the depth of the network. The experimental results show that the proposed model can detect abnormal classes of small samples well on the CICIDS2017 dataset. We not only compare experimentally with other CNN models, but also compare the feature extraction of RNN models on raw traffic. The experimental results show that our models all outperform the other models. Finally, we find that the model has a higher detection accuracy on the payload feature set than the header.

In the future, we will try to use deep learning algorithms to detect unknown types of attacks. In addition, we would like to introduce recurrent neural network models and unsupervised models to mine the temporal and unknown features present in the traffic. Thus, we can improve the detection capability of real-time and persistent attack traffic to keep pace with the development of cyber environment and to improve cyber security in cloud computing scenarios.

Acknowledgments. This research is supported by the National Natura Science Foundation of China under Grant 61873160, Grant 61672338 and Natural Science Foundation of Shanghai under Grant 21ZR1426500.

References

1. Han, D., Pan, N., Li, K.C.: A traceable and revocable ciphertext-policy attribute-based encryption scheme based on privacy protection. IEEE Transactions on Dependable and Secure Computing pp. 1–1 (2020)
2. Cui, M., Han, D., Wang, J.: An efficient and safe road condition monitoring authentication scheme based on fog computing. IEEE Internet of Things Journal 6(5), 9076–9084 (2019)
3. Cui, M., Han, D., Wang, J., Li, K.C., Chang, C.C.: Arfv: An efficient shared data auditing scheme supporting revocation for fog-assisted vehicular ad-hoc networks. IEEE Transactions on Vehicular Technology 69(12), 15815–15827 (2020)
4. Xiao, T., Han, D., He, J., Li, K.C., de Mello, R.F.: Multi-keyword ranked search based on mapping set matching in cloud ciphertext storage system. Connection Science 33(1), 95–112 (2021)
5. Tian, Q., Han, D., Jiang, Y.: Hierarchical authority based weighted attribute encryption scheme. Computer Science and Information Systems 16(3), 797–813 (2019)
6. Kilincer, I.F., Ertam, F., Sengur, A.: Machine learning methods for cyber security intrusion detection: Datasets and comparative study. Computer Networks 188, 107840 (2021)
7. Liu, H., Han, D., Li, D.: Behavior analysis and blockchain based trust management in vanets. Journal of Parallel and Distributed Computing 151, 61–69 (2021)
8. Tian, Q., Han, D., Li, K., Liu, X., Duan, L., Castiglione, A.: An intrusion detection approach based on improved deep belief network. Applied Intelligence 50(10), 3162–3178 (2020)
9. Xu, J., Han, D., Li, K., Jiang, H.: A k-means algorithm based on characteristics of density applied to network intrusion detection. Computer Science and Information Systems 17(2), 665–687 (2020)
10. Sharafaldin., I., Habibi Lashkari., A., Ghorbani., A.A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: Proceedings of the 4th International Conference on Information Systems Security and Privacy - ICISSP,. pp. 108–116. INSTICC, SciTePress (2018)

11. Zhang, Y., Chen, X., Jin, L., Wang, X., Guo, D.: Network intrusion detection: Based on deep hierarchical network and original flow data. IEEE Access 7, 37004–37016 (2019)
12. Jiang, K., Wang, W., Wang, A., Wu, H.: Network intrusion detection combined hybrid sampling with deep hierarchical network. IEEE Access 8, 32464–32476 (2020)
13. Japkowicz, N., Stephen, S.: The class imbalance problem: A systematic study. Intelligent data analysis 6(5), 429–449 (2002)
14. Bailey-Lee, C., Roedel, C., Silenok, E.: Detection and characterization of port scan attacks. Univeristy of California, Department of Computer Science and Engineering pp. 1–7 (2003)
15. Bhuyan, M.H., Kashyap, H.J., Bhattacharyya, D.K., Kalita, J.K.: Detecting distributed denial of service attacks: Methods, tools and future directions. The Computer Journal 57(4), 537–556 (2014)
16. Zhao, G., Xu, K., Xu, L., Wu, B.: Detecting apt malware infections based on malicious dns and traffic analysis. IEEE Access 3, 1132–1142 (2015)
17. Wang, W., Zhu, M., Zeng, X., Ye, X., Sheng, Y.: Malware traffic classification using convolutional neural network for representation learning. In: 2017 International Conference on Information Networking (ICOIN). pp. 712–717 (2017)
18. Maseer, Z.K., Yusof, R., Bahaman, N., Mostafa, S.A., Foozy, C.F.M.: Benchmarking of machine learning for anomaly based intrusion detection systems in the cicids2017 dataset. IEEE Access 9, 22351–22370 (2021)
19. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)
20. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)
21. Marín, G., Caasas, P., Capdehourat, G.: Deepmal-deep learning models for malware traffic detection and classification. In: Data Science–Analytics and Applications, pp. 105–112. Springer (2021)
22. Ran, J., Chen, Y., Li, S.: Three-dimensional convolutional neural network based traffic classification for wireless communications. In: 2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP). pp. 624–627 (2018)
23. Lim, H.K., Kim, J.B., Heo, J.S., Kim, K., Hong, Y.G., Han, Y.H.: Packet-based network traffic classification using deep learning. In: 2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC). pp. 046–051 (2019)
24. Wang, W., Zhu, M., Wang, J., Zeng, X., Yang, Z.: 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 (2017)
25. Zhang, Y., Chen, X., Guo, D., Song, M., Teng, Y., Wang, X.: Pccn: Parallel cross convolutional neural network for abnormal network traffic flows detection in multi-class imbalanced network traffic flows. IEEE Access 7, 119904–119916 (2019)
26. Yujie, P., Weina, N., Xiaosong, Z., Jie, Z., Wu, H., Ruidong, C.: End-to-end android malware classification based on pure traffic images. In: 2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP). pp. 240–245 (2020)
27. Ho, S., Jufout, S.A., Dajani, K., Mozumdar, M.: A novel intrusion detection model for detecting known and innovative cyberattacks using convolutional neural network. IEEE Open Journal of the Computer Society 2, 14–25 (2021)
28. Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A., Lloret, J.: Network traffic classifier with convolutional and recurrent neural networks for internet of things. IEEE Access 5, 18042–18050 (2017)

29. Roopak, M., Yun Tian, G., Chambers, J.: Deep learning models for cyber security in iot networks. In: 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC). pp. 0452–0457 (2019)
30. Feng, J., Shen, L., Chen, Z., Wang, Y., Li, H.: A two-layer deep learning method for android malware detection using network traffic. IEEE Access 8, 125786–125796 (2020)
31. Khan, M.A., Karim, M.R., Kim, Y.: A scalable and hybrid intrusion detection system based on the convolutional-lstm network. Symmetry 11(4) (2019)
32. Shiravi, A., Shiravi, H., Tavallaee, M., Ghorbani, A.A.: Toward developing a systematic approach to generate benchmark datasets for intrusion detection. Computers & Security 31(3), 357–374 (2012)
33. Sun, P., Liu, P., Li, Q., Liu, C., Lu, X., Hao, R., Chen, J.: Dl-ids: Extracting features using cnn-lstm hybrid network for intrusion detection system. Security and Communication Networks 2020 (2020)
34. Wang, M., Zheng, K., Luo, D., Yang, Y., Wang, X.: An encrypted traffic classification framework based on convolutional neural networks and stacked autoencoders. In: 2020 IEEE 6th International Conference on Computer and Communications (ICCC). pp. 634–641 (2020)
35. Zhang, W., Wang, J., Chen, S., Qi, H., Li, K.: A framework for resource-aware online traffic classification using cnn. In: Proceedings of the 14th International Conference on Future Internet Technologies. CFI'19, Association for Computing Machinery, New York, NY, USA (2019)
36. Zhang, C., Chen, Y., Meng, Y., Ruan, F., Chen, R., Li, Y., Yang, Y.: A novel framework design of network intrusion detection based on machine learning techniques. Security and Communication Networks 2021 (2021)
37. NETRESEC: Splitcap (2010), `https://www.netresec.com/index.ashx?page=SplitCap`
38. Chen, Z., He, K., Li, J., Geng, Y.: Seq2img: A sequence-to-image based approach towards ip traffic classification using convolutional neural networks. In: 2017 IEEE International Conference on Big Data (Big Data). pp. 1271–1276 (2017)
39. Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., Chen, T.: Recent advances in convolutional neural networks. Pattern Recognition 77, 354–377 (2018)
40. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Bach, F., Blei, D. (eds.) Proceedings of the 32nd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 37, pp. 448–456. PMLR, Lille, France (07–09 Jul 2015)
41. Azzouni, A., Pujolle, G.: A long short-term memory recurrent neural network framework for network traffic matrix prediction. arXiv preprint arXiv:1705.05690 (2017)
42. Yuan, X., Li, C., Li, X.: Deepdefense: Identifying ddos attack via deep learning. In: 2017 IEEE International Conference on Smart Computing (SMARTCOMP). pp. 1–8 (2017)
43. Hwang, R.H., Peng, M.C., Nguyen, V.L., Chang, Y.L.: An lstm-based deep learning approach for classifying malicious traffic at the packet level. Applied Sciences 9(16) (2019)
44. Kim, A., Park, M., Lee, D.H.: Ai-ids: Application of deep learning to real-time web intrusion detection. IEEE Access 8, 70245–70261 (2020)

Benhui Xia received the B.S. degree from China University of Mining and Technology, where he is currently pursuing the M.S. degree with Shanghai Maritime University. His main research interests include network security, cloud computing, distributed computing and blockchain.

Dezhi Han received the Ph.D. degree from the Huazhong University of Science and Technology. He is currently a Professor of computer science and engineering with Shanghai

Maritime University. His research interests include cloud computing, mobile networking, wireless communication, and cloud security.

Ximing Yin received the M.S. degree from Zhejiang University, where he is currently pursuing the Ph.D. degree with East China University of Science and Technology. His main research interests include network security and wireless network security.

Na Gao received the B.S. degree from Shanxi Agricultural University of Software,where she is currently pursuing the M.S. degree with Shanghai Maritime University. Her main research interests are port supply chain applications and blockchain technology.

Received: June 17, 2021; Accepted: September 18, 2021.