Network Traffic Classification Model Based on Attention Mechanism and Spatiotemporal Features

Feifei Hu  
China Southern Power Grid Co Ltd

Situo Zhang  
China Southern Power Grid Co Ltd

Xubin Lin  
China Southern Power Grid Co Ltd

Liu Wu  
China Southern Power Grid Co Ltd

Niandong Liao (✉ Indy97@csust.edu.cn)  
Changsha University of Science and Technology - Yuntang Campus: Changsha University of Science and Technology

Yanqi Song  
Changsha University of Science and Technology - Yuntang Campus: Changsha University of Science and Technology

Research

Keywords: Traffic Classification, CNN, LSTM, Attention Mechanism

Posted Date: April 1st, 2021

DOI: https://doi.org/10.21203/rs.3.rs-353938/v1

License: ☑️ This work is licensed under a Creative Commons Attribution 4.0 International License. 
Read Full License
Network Traffic Classification Model Based on Attention Mechanism and Spatiotemporal Features

Feifei Hu a, Situo Zhang a, Xubin Lin a, Liu Wu a, Niandong Liao *,b, and Yanqi Song b

aCSG Power, Dispatching Control Center, Guangzhou 510663, China
bChangsha University of Science and Technology, Changsha 410114, China

*Corresponding author.
E-mail addresses:
lndy97@csust.edu.cn(N.Liao),
2078708922@qq.com(Y.Song)
Abstract

Traffic classification has been widely used in network security and network management. Previous research has focused on mapping network traffic to different non-encrypted applications. However, there are few researches on network traffic classification of encryption applications, especially the underlying traffic of encryption application. In order to solve the above problems, this paper proposes a network encrypted traffic classification model which combines attention mechanism with spatial and temporal characteristics. The model first uses LSTM (Long Short-Term Memory) to analyze the time series of the continuous network flows and find out the time characteristics between the network flows. Secondly, CNN (Convolutional Neural Network) is used to extract the high-order spatial features of the network flow, and then the high-order spatial features are weighted and redistributed through the SE (Squeeze-and-Excitation) module to obtain the key spatial features of encrypted traffic. Finally, through the two-stage training and learning, fast classification of network flow is achieved. The main advantages of this model are as follows: 1) the mapping relationship between network flow and corresponding labels is constructed end-to-end without manual extraction of network flow characteristics; 2) it has a powerful generalization ability which is able to be compatible with different types of data sets; 3) there is still a high recognition rate for encryption application and the underlying traffic of encryption application. The experimental results show that this model can be well qualified for the classification of non-encrypted and encrypted application, moreover, greatly improves the classification accuracy of the underlying traffic of encryption application.

Keywords: Traffic Classification, CNN, LSTM, Attention Mechanism
1 Introduction

Network traffic classification is a task that matches network traffic with specific applications or activities. It is of great significance to network management and network security\[1, 2\]. In the field of network management, different types of network applications can be identified by network traffic classification, and proper network resource allocation can be made. In the field of network security, for the intrusion caused by malicious software, network traffic classification can accurately screen the network traffic with malicious behavior and give warnings.

At present, the traditional network traffic classification methods are generally divided into two kinds: port-based and DPI(Deep packet inspection)-based\[3\]. Port-based method identifies network traffic by standard port Numbers. However, in the current network environment, due to the popularity of port obfuscation and dynamic port, the accuracy of port-based method begins to decline. DPI-based method has high classification accuracy for known application traffic, but it cannot identify unknown or encrypt application traffic\[4\].

In recent years, two hot technologies have emerged in the field of traffic classification: classical machine learning and representation learning\[5\]. The former can identify unknown or encrypted application traffic, but relies on prior knowledge. The latter does not need to manually extract the traffic features, it constructs the mapping relationship between network traffic and corresponding tags end-to-end, which saves a lot of labor costs and gets rid of the dependence on prior knowledge on the basis of ensuring effective classification\[6\].

Through the study of previous research achievement\[1\], it can be found that end-to-end construction of the mapping relationship between network flow and corresponding tags is feasible, but the following issues need to be paid attention to:

(1) At present, there is still room for improvement in the accuracy of encrypted network traffic classification\[1, 7, 8, 9\].

(2) Most of the researches have good effects on the classification of specific applications, but they cannot accurately classify the underlying traffic of encryption application\[9\].
In order to solve the above two problems, this paper proposes a self-learning multi-classification network traffic classification model. The model first learns the temporal characteristics on the data set, then, combined with the attention mechanism, which the key feature channels in the feature map are enlarged where learning spatial features, so that the deep neural network can learn the most concerned spatial local information in each group of network traffic. The combination of temporal and spatial information greatly improves the accuracy of classification on the premise of reducing the false positives, especially without consuming too much artificial resources for feature extraction and feature selection.

In this experiment, three data sets were tested to evaluate the performance of the model, and the experimental results were compared with the methods of recent years: [8], [1], [7], [9]. The results show that the model is more accurate and better than the comparison method in different classification experiments.

The main contributions of this paper are as follows:

1. This paper proposes a deep learning network model that automatically classifies network traffic end-to-end.
2. This paper applies the attention mechanism and representation method to network traffic classification to get rid of the bottleneck of information processing and improve the model capability.
3. In this paper, three data sets are used to verify the validity of the classification model. Experimental results show that this method has higher detection accuracy and better generalization ability than other methods.

The rest of this paper is organized as follows. The second section is related work, which introduces the motivation and preparation of the experiment. The third section describes the specific method. The fourth section gives the experimental results and analysis. The fifth section summarizes the paper and prospects the future research.
2 Related work

In real business, traditional methods are still widely used \cite{10,11}. Port-based method uses port numbers to identify applications. For example, the SMTP protocol uses port 25, and TCP uses port 20 or 21. It is an early traffic classification method and very simple to implement. Yeon-sup Lim et al.\cite{12} fully prove that this method can effectively identify applications that follow the port registration rules, but it has a low recognition accuracy for applications with dynamic ports. Nowadays, most scholars do not use the port-based method alone, but mix it with other methods. Chun-Nan Lu et al.\cite{13} further proposed an improved method to classify network traffic based on port and packet length distribution. DPI-based method usually uses patterns or keywords in the payload to identify the application. This method has been developed and has a certain weight in the market. The paper\cite{14} compares two commercial products (PACE and NBAR) and four open source tools (OpenDPI, L7-Filter, nDPI and Libprotoindent). These six tools can solve some network traffic classification problems.

Although these methods have different detection focus, they all need specific rules to identify the corresponding network traffic. However, some new applications no longer follow the port registration rules and use encryption during traffic transmission, which cannot be satisfied by DPI method\cite{15}.

In recent years, some machine learning methods (SVM, Bayes, k-nearest neighbor algorithm and neural network) have been applied in network traffic classification. These methods rely on strong learning ability, and can classify partial encrypted network traffic, but their performance mainly depends on manual feature extraction.

A good data set is a necessary condition to verify the correctness of the method. At present, researchers mainly rely on some well-known network attack datasets (such as KDDCUP99\cite{16}, NSL-KDD\cite{17}, UNSW-NB15\cite{18}) to verify the proposed method. Gao et al.\cite{19} tried to combine multiple boltzmann machines and back propagation algorithm to carry out pre-training and fine-tuning of the network, and this method achieved good results on KDDCUP99 data set. Javaid et al.\cite{20} developed a network intrusion detection system, which can be used for unpredictable attacks, and the system has been effectively verified on NSL-KDD data set. Zhang et al.\cite{21} proposed an intrusion detection method combining
multi-scale CNN and LSTM, and tested it on UNSW-NB15 data set. This method can observe spatial information on different scales, and has the ability of fast optimization.

Because machine learning relies too much on the correct selection of feature set for traffic classification, it can not adapt to the new changes of network flow. However, deep learning has a strong self-learning ability and does not need personnel intervention. Deep learning can reduce the labor consumption of manual feature extraction by transforming different types of traffic features into images[1].

The method based on deep learning abandons the previous tedious steps of manual feature extraction and realizes the end-to-end training from the original packet to the classification result.

Deep learning takes the network traffic bytes as the pixels in the image, and learns the shallow and deep features of network traffic from the perspective of computer vision.

At present, the main popular deep learning models are CNN, RNN and so on. These models have made great achievements in computer vision[22], natural language processing[23, 24], speech recognition [25, 26] and other fields.

Wang[7] et al. proposed a malware traffic classification method based on representation learning, and proved the feasibility of representation learning in traffic classification, but this paper did not demonstrate the parameter optimization and generalization ability of CNN model.

Li[27] et al. used RNN and attention mechanism to fully extract the temporal features of the network stream. In order to verify the feasibility of the proposed method, different types of network flows are selected for testing and verification. From the test results, the overall detection rate of the method is greatly improved compared with the machine learning method.

In the research of network flow, using a single deep learning method (such as RNN, CNN) compared with multiple combination methods (such as RNN + LSTM, CNN + LSTM), most scholars found that the combination method has certain advantages through experiments. The combination method can automatically mine the temporal and spatial characteristics of network traffic from multiple perspectives, and reduce the error rate of classification[28].

At present, the identification of encrypted traffic has become an unavoidable problem
Mohammad et al. [9] tried to use two deep learning methods of stackable automatic encoder and CNN to identify encrypted traffic and non-encrypted traffic. This method has been tested in ISCX network flow, and the test results show that it has good detection effect on tor, youtube and other widely known types, but it has not solved the classification problem of these flow subtypes.

From the above research, we can see that the complex network integrates a variety of different network functions and services, and these services contain a large number of encrypted and non-encrypted data. How to quickly find abnormal flow from network flow is still a challenging scientific problem, which needs continuous exploration and efforts.
3 Proposed solution

In order to further improve the classification performance of deep learning in network flow, CNN is used to extract the spatial features of network flow, and LSTM is used to extract the temporal features of network flow.

In this section, a deep learning method combining temporal information with spatial information is designed. The overall network architecture is shown in figure 1.

The detection model combines the advantages of LSTM and CNN model, which can effectively learn the representation of feature graph and avoid the manual design of features by experts. The main ideas are as follows:

1. Firstly, LSTM is used to extract the temporal features of network flow image;
2. Then, Squeeze-and-Excitation (SE) mechanism is used to optimize the CNN network structure and improve the training performance of CNN.

LSTM is kind of neural network with memory function and a variant of recurrent neural network (RNN). LSTM has excellent performance in temporal data processing. Network flow is a typical time series data, so it is very suitable for LSTM training.

CNN model has a strong ability of image feature extraction, which usually includes three main components: convolution layer, pooling layer and full connection layer. The convolution layer works by convoluting the local region of the input data with the convolution kernel. The function of pooling layer is to reduce the dimension of training characteristics. Fully connected layer is a traditional multi-layer perceptron, which is often used as output.

CNN-SE-net refers to combining the SE module on the traditional CNN network. The SE module is an attention mechanism that can be embedded in other classification or detection models. The core idea is to learn the feature weights according to the loss function through the network, so that the effective feature map has a larger weight, and the invalid or less effective feature map has a smaller weight.

After these two steps of optimization, the training model can achieve better results. The implementation of the model is in Chapter 3.3.
3.1 Datasets

This paper uses three common data sets, ISCX VPN-nonVPN[29], USTC-TFC2016[30] and YouTube dataset[31]. These three datasets contain a lot of encrypted, unencrypted, abnormal and normal traffic.

ISCX dataset is saved in pcap format from Canadian network security research institute. Pcap files are a common datagram storage format. As shown in table 1, according to the protocol type, the data set divides the network traffic into 12 types (chat, e-mail, file transfer, streaming, torrent, VoIP).

USTC-TFC dataset comes from CTU research, which is also in pcap format. As shown in table 1, the dataset contains 20 types of network flows, including 10 normal and abnormal flows. Normal traffic is simulated by Ixia BPS[32] and abnormal traffic is collected by researchers in real network environment.

The YouTube dataset is a collection of 100 encrypted video streams from chrome. The titles of these videos come from current hot areas, such as news, sports, nature, etc.[33]. Because the data set is too large and various, this paper selects 10 samples as the training set and test set.
Table 1. Labels for Three Datasets

<table>
<thead>
<tr>
<th>ISCX VPN-nonVPN</th>
<th>Traffic</th>
<th>Malware Traffic</th>
<th>Normal Traffic</th>
<th>YouTube</th>
</tr>
</thead>
<tbody>
<tr>
<td>Email</td>
<td>VPN-Email</td>
<td>Cridex</td>
<td>BitTorrent</td>
<td>American_Hustle</td>
</tr>
<tr>
<td>Chat</td>
<td>VPN-Chat</td>
<td>Geodo</td>
<td>Facetime</td>
<td>BonBon</td>
</tr>
<tr>
<td>Streaming</td>
<td>VPN-Streaming</td>
<td>Htbot</td>
<td>FTP</td>
<td>Disconnect</td>
</tr>
<tr>
<td>File transfer</td>
<td>VPN-File transfer</td>
<td>Miuref</td>
<td>Gmail</td>
<td>Friends</td>
</tr>
<tr>
<td>VoIP</td>
<td>VPN-VoIP</td>
<td>Neris</td>
<td>MySQL</td>
<td>Hollyweezy</td>
</tr>
<tr>
<td>P2P</td>
<td>VPN-P2P</td>
<td>Nsis-ay</td>
<td>Outlook</td>
<td>Let_It_Go</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Shifu</td>
<td>Skype</td>
<td>Maria</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tinba</td>
<td>SMB</td>
<td>Maroon_5_Sugar</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Virut</td>
<td>Weibo</td>
<td>Sola</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Zeus</td>
<td>WorldOfWarcraft</td>
<td>TenYears</td>
</tr>
</tbody>
</table>

3.2 Preprocessing

Since the original network packet in pcap form cannot be directly used as the input of this model, this paper pre-processes the three datasets in four steps: traffic filtering, image generation, and IDX conversion.

Step 1 (traffic filtering): since the session is a bidirectional network flow, which contains more information than the one-way flow, this paper decides to adopt the traffic classification method based on the session mode. This step splits the raw packets in pcap format into a single session-level packet and then cleans the traffic, removing the empty and duplicate files that affect the model’s training.

Step 2 (image generation): in order to visualize the data, this step unites all pcap files into 784 bytes in length. Larger than 784 bytes is clipped, and smaller than 784 bytes is filled with 0. Each byte in the original packet represents one pixel of images, which constitutes the image matrix (28x28x1), that is, each images has three parameters: high H, wide W, and channel C. Some single-channel images as shown in the figure 2, figure 3 and figure 4.

Step 3 (IDX format conversion): The data is processed into IDX file format as the input of CNN network.
Fig. 2. Visualization of ISCX VPN-nonVPN

Fig. 3. Visualization of STC-TFC2016
3.3 Model

3.3.1 LSTM

Network traffic is transmitted in the form of sequence in the network, which contains rich time-related information. In order to analyze network traffic comprehensively and extract time information, LSTM model is adopted in this paper. LSTM is more complex and sophisticated in information processing than RNN, which can solve the problem of RNN in remote information processing.

In this paper, the preprocessed data is transformed into a single channel image, and then these features are mapped into the classic LSTM model. Through LSTM model training and learning, the time series features of network flow are accurately extracted.

The internal structure of LSTM is relatively complex[26], and the core is the unit state flow. The unit state flow is controlled and adjusted by three gate mechanisms: the forgotten gate $F_t$, the input gate $I_t$, and the output gate $O_t$.

The output of the previous cell $h_{t-1}$ and the input data of the current cell $x_t$ are entered into the forgetting gate at the same time to obtain the information retention degree of the previous hidden layer, $F_t$.

$$F_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$$  \hspace{1cm} (1)

Where $\sigma$ refers to the sigmoid function, $W_f$ is the weight matrix of the forgotten gate, and
$b_f$ is the bias term. The input gate calculates $I_t$ and $C'_t$ to determine the new data and the extent to which it needs to be retained. The above is then combined with the output $F_t$ of the forgotten gate and the previous cell state $C_{t-1}$ to update the current cell state $C_t$.

$$I_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$$  
\hspace{1cm} (2)

$$C'_t = tanh(W_c[h_{t-1}, x_t] + b_c)$$  
\hspace{1cm} (3)

$$C_t = F_tC_{t-1} + I_tC'_t$$  
\hspace{1cm} (4)

where $W_i$ and $W_c$ are the weight matrix of the input gate, and $b_i$ and $b_c$ are the bias term. Finally, the output gate integrates the output of the above two doors to get the final update result $h_t$, and $O_t$ determines which part of the cell state we want to output.

$$O_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$  
\hspace{1cm} (5)

$$h_t = O_t tanh(C_t)$$  
\hspace{1cm} (6)

Where $W_o$ is the weight matrix of the output gate, and $b_o$ is the bias term.

### 3.3.2 CNN-SE-net

As an excellent deep learning image processing model, CNN uses shared convolution kernel to process high-dimensional data without pressure, and can automatically extract features.

The researchers also found that a single CNN network also has the following problems: (1) When the network level is too deep, using BP propagation to modify the parameters will make the parameter changes near the input layer slower; (2) It is easy to make the training result converge to the local minimum instead of the global minimum by using the gradient descent algorithm; (3) The pooling layer will lose a lot of valuable information, ignoring the correlation between the part and the whole.

Therefore, it is not possible to directly use a single CNN model when performing network flow training. In order to improve the accuracy and efficiency of the CNN model in network traffic classification, this paper has optimized settings from two aspects.
First of all, during the training of the CNN model, the perceptual field of the large convolution kernel can be simulated by stacking 3x3 convolution kernels. For example, three 3x3 convolution kernels are stacked, and the receiving field is equal to the receiving field of the 7x7 convolution.

This method can increase the depth and search space of the network, reduce the number of model parameters, and improve the performance of the model [34]. In addition, during model training, choosing a smaller slide can prevent the loss of detailed information due to a larger slide. Therefore, this paper sets the model slide parameter to 1.

Secondly, this paper uses the self-attention mechanism of the SE module [35] to extract spatial information and channel information, and relocate the interdependence of the feature map channels. The SE module generates the modulation weight of the global information of the feature map, and enhances or suppresses different channels for different classification schemes.

Attention mechanism is also known as ”attention of neural network”. Attention mechanism can be divided into three steps: first, information input; second, calculating attention distribution; third, processing input information according to the calculated attention distribution.

Let $a \in \mathbb{R}^d$ be the input vector, $X = [x_1, x_2, \cdots, x_n]$ be N input samples, $q \in \mathbb{R}^k$ be the query vector or feature vector, and $Z \in [1, N]$ be the attention variable, which indicates the position of the selected information. For example, $z = i$ means the $i$-th input vector is selected.

The general attention mechanism is divided into soft attention and hard attention. The formulas of soft attention mechanism is generally as follow:

$$a_i = p(z = i \parallel X, q) = \text{softmax}(s(x_i, q))$$

(7)

Where $a_i$ is called attention distribution, $s(x_i, q)$ is the attention scoring function. Attention distribution $a_i$ can be interpreted as the degree of attention of the $i$-th input vector for a given query $q$. The soft attention selection mechanism is to aggregate them.

$$\text{att}(X, q) = \sum_{i=1}^{n} a_i X_i$$

(8)
Hard attention selects information based on maximum sampling or random sampling. Among them, the formula for selecting the input information with the highest probability is:

$$\text{att}(X, q) = X_j, \text{where } j = \arg \max_{i=1}^{N} a_i$$  \hspace{1cm} (9)

The specific implementation of embedding SE in CNN is shown in Figure 5. After preprocessing, the original network flow is input to LSTM module for time series analysis, and the output $X = [x_1, x_2, ..., x_c]$ is obtained. After the convolution operation $F_{tr}$, the output is $U' = [u_1', u_2', ..., u_c'], U' \in R^{H' \times W' \times C'}$.

The convolution operation is shown in formula (10).

$$U' = X \ast V_k, (k = 1, 2, ..., C)$$  \hspace{1cm} (10)

Where $\ast$ denotes convolution and $V = [v_1, v_2, ..., v_c]$ is a convolution kernel containing $C$ 3x3. The first three convolution layers of network use 32, 64 and 64 convolution kernels respectively, and the channel number of the feature graph is correspondingly converted to: 1-32-64-64.

After that, the fourth network layer is the maximum pooling layer with 2x2 and step size of 2, and the output result is $U = [u_1, u_2, ..., u_c], U \in R^{H \times W \times C}$.

This paper embeds the SE module in the fifth layer of the network, and specifically refers to the specific operation of the SE in the literature [35]. This SE is a simple and computationally small channel-based attention model.

The SE mechanism uses weights to enhance or suppress feature channels, so that the model focuses on more important spatial feature information. The specific operation is further subdivided into three main steps: global average pooling $F_{sq}(\cdot)$, weight generation $F_{ex}(\cdot)$ and weight redistribution $F_{sca}(\cdot)$. 
Fig. 5. SE mechanism embedded flow chart

(1) Global average pooling

The feature map output by the maximum pooling layer has three dimensions, namely width, height, and number of channels. First, the global average pooling operation compresses the width and height directions, so that the width and height dimensions are reduced to 1x1, but the number of channels remains unchanged.

The feature map output in this step is $W^* = [w_1^*, w_2^*, ..., w_C^*]$, $W_1^* \in \mathbb{R}^{1 \times 1 \times C}$. The feature map of 1x1xC has a global perceptual domain, and the specific calculation of $W^*$ is shown in formula (11):

$$W^* = F_{sq}(U) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_{ij}, (k = 1, 2, \ldots, C) \quad (11)$$
(2) Weight generation

Two fully connected layers are used to reduce and increase the dimension of channel C, increase the non-linear relationship between channels. This operation produces a weight representing the importance of a set of global information \( W^\sim = [w_1^\sim, w_2^\sim, ..., w_C^\sim] \). The details are described in the formula (12).

\[
W^\sim = F_{ex}(W^*, W) = \sigma(g(W^*, W)) = \sigma(W_2\sigma_2(W_1W^*))
\]  

where \( W_1 \in R^{C \times C}, W_2 \in R^{C \times R}, r \) is reduction ratio of dimensionality-reduction layer.

(3) Weight redistribution

Finally, the output \( U \in R^{H \times W \times C} \) of the maximum pooling layer is multiplied by the feature channel weight \( W^\sim = [w_1^\sim, w_2^\sim, ..., w_c^\sim] \) generated by the previous operation, and the output of CNN-SE-net \( X^\sim = [x_1^\sim, x_2^\sim, ..., x_c^\sim] \) is obtained.

\[
x_c^\sim = F_{sea}(u_c, w_c^\sim) = w_c^\sim u_c
\]

It can be seen that the importance of each channel has changed after weight redistribution.

The output processed by SE module passes through three convolution layer of the small convolution kernel of 3x3 again. The three convolution layers respectively use 32, 32 and 16 convolution kernels, and the channel number is converted to: 32-32-16. Then the model is trained in a maximum pooling layer with a dimension of 2x2 and a step size of 2, a fully connected layer with a dimension of 1024 and a fully connected layer with a dimension of traffic, and finally the softmax layer outputs the final classification results.
4 Experiment

4.1 Basic performance test

In the experiment of this paper, we selected three internationally network public traffic data sets: ISCX VPN-nonVPN, USTC-TFC2016 and YouTube video data sets. The data types of these data sets are all pcap files, and the specific introduction of the data sets is in chapter 3.1. After all the data is preprocessed as described in section 3.2, each network flow is cut to form a 28x28 image. The parameter configuration set in the experiment is shown in Table 2.

<table>
<thead>
<tr>
<th>parameter</th>
<th>values</th>
</tr>
</thead>
<tbody>
<tr>
<td>learning rate</td>
<td>$10^{-4}$</td>
</tr>
<tr>
<td>Number of training(12 classification)</td>
<td>$3.5 \times 10^5$</td>
</tr>
<tr>
<td>Number of training(20 classification)</td>
<td>$6.5 \times 10^5$</td>
</tr>
<tr>
<td>batch(12 classification)</td>
<td>64</td>
</tr>
<tr>
<td>batch(20 classification)</td>
<td>256</td>
</tr>
<tr>
<td>Time stamp of LSTM</td>
<td>28</td>
</tr>
</tbody>
</table>

In order to evaluate the proposed detection scheme, this paper uses Python, Scikit-learn, NumPy, Pandas, TensorFlow and Keras machine learning libraries. First, we randomly selected 90% training set samples and 10% test set samples from the data sets. To prevent overfitting problems, add the dropout layer after the first full connection layer. The loss functions and optimizers used are crossentropy and Adam, respectively, and mainly use ReLU and softmax two excitation functions in training phase.

In order to evaluate the effect of this method and have an objective comparison with other methods, this paper uses F1-Score, recall, accuracy and precision to evaluate the experimental results.

$$\text{accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$  \hspace{1cm} (14)
\[ \text{precision} = \frac{TP}{TP + FP} \]  
(15)

\[ \text{recall} = \frac{TP}{TP + FN} \]  
(16)

\[ F1 \text{ Score} = \frac{2 * \text{precision}}{\text{precision} + \text{recall}} \]  
(17)

In the formula, \( TP \) (True Positive): The number of successful detections of the current network traffic category. \( TN \) (True Negative): The number of other network traffic types successfully detected. \( FP \) (false positive): The number of other network traffic categories identified as the current network traffic category. \( FN \) (False Negative): The number of current network traffic categories identified as other network traffic categories.

The performance results obtained from the experiment are shown in Figures 6, figure 7 and figure 8. In these figures, the value on the X axis represents the training time of the training set samples in the model, the loss on the Y axis represents the loss value of the model training, and accuracy represents the detection accuracy of the model, and these two values are respectively red and blue curve to mark.

From the experimental results of the three figures, it can be observed that as the model training epoch continues to increase, the detection accuracy of the model is getting higher and higher, constantly approaching 100%, and the loss rate is also getting lower and lower, gradually approaching 0%.

In addition, it can be seen from the three figures that the training time of the model in different data sets does not require a long time, and basically it can reach the optimal value of classification in about 200 seconds. The reduction in training time is conducive to reducing the consumption of computing resources.
Fig. 6. Test performance on ISCX VPN-nonVPN

Fig. 7. Test performance on USTC-TFC2016
In order to further verify the overall performance of the model on different data sets, especially to determine whether different network traffic types can be accurately classified, this paper has completed training and testing on three data sets.

The test results obtained in the experiment are shown in Tables 3, table 4 and table 5. From the experimental results of the three data sets, the values of Precision, F1-score and recall are all very good. Specifically, each type of network traffic can be accurately classified in the model, and the accuracy is more than 93%. In addition, in Table 3, the average accuracy has reached 97.54%, the average F1 score is 97.61%, and the average recall is 97.72%. In tables 4 and 5, the precision, F1 score and recall are close to 99%.

From the detection results of these three tables, it fully shows that the proposed combination of LSTM and CNN for network traffic classification is completely feasible. The model can better extract the time and space characteristic information of network traffic, and then use the attention mechanism to optimize the feature selection, which further improves the learning ability of the model.
<table>
<thead>
<tr>
<th>category</th>
<th>Recall($10^{-2}$)</th>
<th>Precision($10^{-2}$)</th>
<th>F1-Score($10^{-2}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chat</td>
<td>97.01</td>
<td>99.84</td>
<td>98.40</td>
</tr>
<tr>
<td>Email</td>
<td>100.00</td>
<td>96.91</td>
<td>98.43</td>
</tr>
<tr>
<td>File</td>
<td>96.80</td>
<td>92.13</td>
<td>94.40</td>
</tr>
<tr>
<td>P2p</td>
<td>100.00</td>
<td>99.05</td>
<td>99.52</td>
</tr>
<tr>
<td>Streaming</td>
<td>98.92</td>
<td>99.46</td>
<td>99.19</td>
</tr>
<tr>
<td>Voip</td>
<td>92.06</td>
<td>96.98</td>
<td>94.45</td>
</tr>
<tr>
<td>Vpn_Chat</td>
<td>99.75</td>
<td>99.86</td>
<td>99.75</td>
</tr>
<tr>
<td>Vpn_Email</td>
<td>93.33</td>
<td>90.32</td>
<td>91.80</td>
</tr>
<tr>
<td>Vpn_File</td>
<td>97.00</td>
<td>97.98</td>
<td>97.49</td>
</tr>
<tr>
<td>Vpn_P2p</td>
<td>97.92</td>
<td>97.92</td>
<td>97.92</td>
</tr>
<tr>
<td>Vpn_Streaming</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td>Vpn_Voip</td>
<td>99.83</td>
<td>100.00</td>
<td>99.92</td>
</tr>
</tbody>
</table>
Table 4. Classification effect of the model on USTC-TFC2016

<table>
<thead>
<tr>
<th>category</th>
<th>Recall(10^{-2})</th>
<th>Precision(10^{-2})</th>
<th>F1-Score(10^{-2})</th>
</tr>
</thead>
<tbody>
<tr>
<td>BitTorrent</td>
<td>100.00</td>
<td>99.88</td>
<td>99.94</td>
</tr>
<tr>
<td>Facetime</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td>FTP</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td>Gmail</td>
<td>99.53</td>
<td>99.88</td>
<td>99.71</td>
</tr>
<tr>
<td>MySQL</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td>Outlook</td>
<td>99.87</td>
<td>99.47</td>
<td>99.67</td>
</tr>
<tr>
<td>Skype</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td>SMB</td>
<td>99.68</td>
<td>99.84</td>
<td>99.76</td>
</tr>
<tr>
<td>Weibo</td>
<td>99.85</td>
<td>99.85</td>
<td>99.85</td>
</tr>
<tr>
<td>WorldOfWarcraft</td>
<td>100.00</td>
<td>99.87</td>
<td>99.93</td>
</tr>
<tr>
<td>Cridex</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td>Geodo</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td>Htbot</td>
<td>99.84</td>
<td>100.00</td>
<td>99.92</td>
</tr>
<tr>
<td>Miuref</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td>Neris</td>
<td>93.05</td>
<td>98.50</td>
<td>95.67</td>
</tr>
<tr>
<td>Nsis-ay</td>
<td>99.17</td>
<td>99.50</td>
<td>99.33</td>
</tr>
<tr>
<td>Shifu</td>
<td>100.00</td>
<td>99.86</td>
<td>99.95</td>
</tr>
<tr>
<td>Tinba</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td>Virut</td>
<td>97.85</td>
<td>91.17</td>
<td>94.36</td>
</tr>
<tr>
<td>Zeus</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
</tr>
</tbody>
</table>
Table 5. The classification effect of the model on YouTube

<table>
<thead>
<tr>
<th>category</th>
<th>Recall($10^{-2}$)</th>
<th>Precision($10^{-2}$)</th>
<th>F1-Score($10^{-2}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>American_Hustle</td>
<td>100.00</td>
<td>99.98</td>
<td>99.94</td>
</tr>
<tr>
<td>BonBon</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td>Disconnect</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td>Friends</td>
<td>99.93</td>
<td>99.98</td>
<td>99.91</td>
</tr>
<tr>
<td>Hollyweezy</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td>Let_It_Go</td>
<td>99.97</td>
<td>99.97</td>
<td>99.97</td>
</tr>
<tr>
<td>Maria</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td>Maroon_5_Sugar</td>
<td>99.98</td>
<td>99.94</td>
<td>99.96</td>
</tr>
<tr>
<td>TenYears</td>
<td>99.95</td>
<td>99.95</td>
<td>99.95</td>
</tr>
</tbody>
</table>

4.2 Comparative Experiment

In order to further verify the performance of the proposed model, this paper selects several methods for comparative experiments. These methods are basically experiments done on the first three data sets.

Gerard Draper-Gil et al. [8] used the well-known machine learning technique C4.5 to test the traffic classification in the ISCX VPN-nonVPN dataset. Wang et al. [1] proposed the application of one-dimensional convolutional neural network to traffic classification. Compared with the C4.5 machine method, the performance of this method is improved by about 5In addition, in order to solve the classification problem of encrypted VPN traffic, this paper uses Multichannel LeNet-5[7] method. This method is similar to the LeNet-5 network structure, which improves detection performance by setting more channels in each convolutional layer. Mohammad Lotfollahi et al. [9] used a combination of SAE and 1D-CNN to classify network flows. Literature [33] uses traditional machine learning K-nearest neighbor method to classify network flows.

Under the same training data set and test data set, the completed comparative experiment results are shown in Tables 6, 7, and 8.

In Table 3 the detection results of the ISCX VPN-nonVPN data set, the C4.5 method
has the lowest detection performance, and the detection effect of the two CNN models is greatly improved. For example, the Precision value of C4.5 is only 78.2%, while the Precision value of SAE+1DCNN reaches 97.8%. At the same time, we also found that the overall performance of the method proposed in this paper is better than the previous three methods in the classification and detection of VPN and Non-VPN flows.

From the experimental results of Table 7 USTC-TFC2016, the multi-channel LeNet-5 network can basically meet the requirements for two different network flow classifications. For example, the detection rate in Benign has reached 97%, and the detection rate in Malware has exceeded 98%, the overall effect is good. Like Multichannel LeNet-5, the method proposed in this paper performs very well in the classification and detection of Benign and Malware streams, and all indicators basically exceed 98%.

In Table 8, this paper only selects the precision index value and compares it with the KNN method. The precision of these two methods is very high on YouTube, especially the method in this paper reaches 99%.

**Table 6.** The test results of ISCXVPN-nonVPN datasets

| Method       | VPN              | Non-VPN          |               |               |               |               |               |
|--------------|------------------|------------------|---------------|---------------|---------------|---------------|
|              | Precision        | Recall           | F1-Score      | Precision     | Recall        | F1-Score      |
| C4.5[8]      | 78.2%            | 81.3%            | -             | 84.3%         | 79.3%         | -             |
| 1D CNN [1]   | 92%              | 95.2%            | -             | 85.8%         | 85.9%         | -             |
| SAE+1D CNN[9]| 97.8%            | 96.3%            | 97%           | 86.7%         | 88.8%         | 87.3%         |
| Our          | **99.4%**        | **99.4%**        | **99.4%**     | **97.4%**     | **97.5%**     | **96.8%**     |

**Table 7.** The test results of USTC-TFC2016 datasets

<table>
<thead>
<tr>
<th>Method</th>
<th>Benign</th>
<th></th>
<th>Malware</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
<td>F1-Score</td>
<td>Precision</td>
<td>Recall</td>
<td>F1-Score</td>
</tr>
<tr>
<td>Multichannel LeNet-5[7]</td>
<td>99.7%</td>
<td>99.7%</td>
<td>99.7%</td>
<td>98.6%</td>
<td>98.7%</td>
<td>98.6%</td>
</tr>
<tr>
<td>Our</td>
<td><strong>99.8%</strong></td>
<td><strong>99.8%</strong></td>
<td><strong>98.9%</strong></td>
<td><strong>98.9%</strong></td>
<td><strong>98.9%</strong></td>
<td><strong>98.9%</strong></td>
</tr>
</tbody>
</table>
Table 8. The test results of Youtube datasets

<table>
<thead>
<tr>
<th>Method</th>
<th>K-Nearest Neighbor Algorithm [33]</th>
<th>Our</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>95.0%</td>
<td>99.9%</td>
</tr>
</tbody>
</table>
5 Conclusion

This paper proposes a deep learning traffic model based on the combination of LSTM, CNN and SE in order to solve the problem that the existing network traffic types are diverse, encrypted and non-encrypted mixed and implicit traffic sub-types are difficult to classify in time. This method first eliminates the problem that some traditional machine learning methods rely too much on the accurate extraction of network traffic characteristics. Secondly, by using the LSTM method to automatically obtain the time series characteristics of the network flow and the CNN method to obtain the spatial characteristics of the flow, the problem of the correlation of the flow characteristics and the incompleteness of the characteristics is better solved. In addition, by embedding the SE mechanism in the CNN, the correlation of the channels between different layers of the network is further analyzed to improve the accuracy of model feature selection. Judging from the results of different experiments, it fully reflects that the method proposed in this paper is indeed feasible and can basically handle the classification of different network traffic.

Because of the channel-based attention mechanism method currently used in this paper, only the global characteristics of the channel are learned, and the global information in time and space has not been further analyzed. The next step is to focus on continuing to optimize the deep learning network structure. In addition, in the SE mechanism, the weight of the temporal and spatial characteristics of the traffic is analyzed to find more reasonable network traffic characteristic data, which lays the foundation for improving the efficiency of model training.

abbreviations and acronyms

The following table describes the significance of various abbreviations and acronyms used throughout the thesis. The page on which each one is defined or first used is also given. Nonstandard acronyms that are used in some places to abbreviate the names of certain white matter structures are not in this list.
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Means</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM</td>
<td>Long Short-Term Memory, which is a kind of time cycle neural network</td>
<td>2</td>
</tr>
<tr>
<td>CNN</td>
<td>Convolutional Neural Network, which is a kind of feedforward neural network with convolution calculation and deep structure</td>
<td>2</td>
</tr>
<tr>
<td>SE</td>
<td>Squeeze and Excitation, its purpose is to improve the quality of the representation generated by the network by explicitly modeling the interdependence between the channels of its convolution features.</td>
<td>2</td>
</tr>
<tr>
<td>DPI</td>
<td>DPI is called &quot;Deep Packet Inspection&quot;. DPI technology adds analysis to the application layer on the basis of analyzing the packet header. It is a flow detection and control technology based on the application layer.</td>
<td>3</td>
</tr>
<tr>
<td>SMTP</td>
<td>SMTP is a protocol that provides reliable and effective email transmission.</td>
<td>5</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol, it is a connection-oriented, reliable, byte stream-based transmission layer communication protocol.</td>
<td>5</td>
</tr>
<tr>
<td>SVM</td>
<td>Support Vector Machine, which is a generalized linear classifier that performs binary classification of data in a supervised learning method.</td>
<td>5</td>
</tr>
<tr>
<td>Bayes</td>
<td>Bayesian classification algorithm is a classification method of statistics.</td>
<td>5</td>
</tr>
<tr>
<td>RNN</td>
<td>Recurrent Neural Network, it is a type of recurrent neural network that takes sequence data as input, recursively in the evolution direction of the sequence, and all nodes (cyclic units) are connected in a chain.</td>
<td>6</td>
</tr>
<tr>
<td>IDX</td>
<td>IDX is an image format</td>
<td>10</td>
</tr>
<tr>
<td>CNN-SE-ne</td>
<td>CNN-SE-net refers to the SE mechanism embedded in the CNN network.</td>
<td>13</td>
</tr>
<tr>
<td>BP</td>
<td>Back propagation, which is a multi-layer feedforward neural network trained according to the error back propagation algorithm.</td>
<td>13</td>
</tr>
<tr>
<td>C4.5</td>
<td>Which is an algorithm developed by Ross Quinlan for generating decision trees.</td>
<td>24</td>
</tr>
<tr>
<td>VPN</td>
<td>Virtual private network</td>
<td>24</td>
</tr>
<tr>
<td>1D-CNN</td>
<td>Which refers to a one-dimensional CNN network.</td>
<td>24</td>
</tr>
<tr>
<td>SAE</td>
<td>Stacked AutoEncoders</td>
<td>25</td>
</tr>
</tbody>
</table>
Acknowledgements

This work is supported by the key scientific and technological project of "Research and Application of Key Technologies for Network Security Situational Awareness of Electric Power Monitoring System (No. ZDKJXM20170002)” of China Southern Power Grid Corporation; Open fund project of the Hunan provincial Engineering Research Center of Electric Transportation and Smart Distribution Network (No. 3040102-1105008).

Authors’ contributions

FH, SZ and XL contributed to the design and implementation of the study and writing part of the paper. NL and YS conducted analysis and simulation experiments and LW supplemented the manuscript.

Authors’ information

Hu Feifei is an employee of power dispatching and control center of China Southern Power Grid Corporation. His main research interests include power system security, network situation awareness and power data security. Email: huff@csg.cn

Funding

There is no financial support for this study.

Availability of data and materials

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.
Competing interests

The authors declare that they have no competing interests.
References


[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-

Figures

Figure 1
network architecture

Figure 2
Visualization of ISCX VPNnonVPN
Figure 3
Visualization of STCTFC2016

Figure 4
Visualization of YouTube
Figure 5

SE mechanism embedded flow chart
Figure 6

Test performance on ISCX VPNnonVPN
Figure 7

Test performance on USTCTFC2016
Figure 8

Test performance on YouTube