Classification method of IPv6 traffic based on convolutional neural network

Pan Ying, Wang Shihui* and Xu Xing
Information Network Center, Guangxi University, Nanning Guangxi 530004, China
*Corresponding author’s e-mail: shwang@gxu.edu.cn

Abstract. This paper proposes a method of IPv6 traffic classification based on convolutional neural network. It aims to solve the problems of low accuracy, high dependence on traffic feature selection, inefficiency in IPv6 traffic classification technology based on machine learning, and the lack of research on IPv6 traffic classification methods. The method transforms IPv6 traffic into two-dimensional matrix as input of convolution neural network. Then, it constructs a traffic classification model of feature autonomous learning. The model consist of one input layer, two convolutional pooling layers, one full connection layer and one output layer. The experimental results show that the designed model has a good performance in IPv6 traffic classification.

1. Introduction

With the rapid growth of Internet users and network devices, IPv4 address resources are gradually exhausted. Then the Internet Engineering Task Force (IETF) proposed the next generation of Internet protocol IPv6 to solve the problem. Then the analysis of IPv6 traffic has gradually become a research hotspot [1]. Network traffic classification is the mapping of network IP messages to an application category, such as Web traffic, YouTube videos, etc. Research on network traffic classification can help to network optimization, QoS, network planning, and user behavior analysis, etc. [2].

At present, there are mainly three network traffic classification methods. One method is port-based traffic classification method, which is simple and efficient. However, many applications use camouflage or non-standardized registration port numbers, then the port-based method gradually becomes ineffective [3]. Another traffic classification method is deep packet inspection (DPI) based on packet load [4], which can solve the problem caused by using camouflaged or non-standardized registration port numbers. However, the method need to detect packet load content, which will involve user privacy issues. There is a method based on statistical features which usually adopts classical machine learning algorithm, including unsupervised clustering algorithm and supervised classification algorithm, and achieves high classification accuracy [5-7]. However, the accuracy of method highly depends on the selection of traffic features, and the algorithm is time consuming.

Considering the deep learning algorithm automatically selects features in the training process, then the algorithm can be used to solve the problem of feature selection dependence in the traffic classification method based on machine learning algorithm. Reference [8] constructs an end-to-end traffic classification model using a traditional one-dimensional convolutional neural network (CNN). It processes the data into a specific file form and then constructs feature space to achieve good classification results. Reference [9] designed a convolutional neural network model, which can autonomous learn traffic features through the convolutional layer, to solve the problem of feature selection in the traditional traffic classification algorithm based on machine learning. Reference [10]
designed a model based on LeNet-5 convolution neural network. It also can autonomous learn traffic features, then to improve the accuracy of the traffic classification.

At present, most traffic classification research is for IPv4 traffic. And the traffic classification accuracy highly depend on traffic feature selection. The paper proposed a classification method of IPv6 traffic based on convolutional neural network. It firstly constructs a CNN model which suitable to IPv6 traffic classification, and then processes the IPv6 traffic to two-dimensional matrix form as the input of the CNN traffic model. The method can autonomous learn traffic features from the training data and extract the traffic feature. It solves the artificial feature selection and improve the classification accuracy.

The rest of this paper is organized as follows. The next section presents a review of deep learning and convolutional neural network. Section 2 explains the proposed method based on convolutional neural network for IPv6 traffic classification. Section 3 presents the experimental results to analyze and evaluate the accuracy of the proposed traffic classification model, and Section 4 concludes the paper.

2. Deep Learning and Convolutional Neural Network

Deep Learning is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks. There are researches from different fields consider using deep learning framework in their areas of research. Many deep learning models, such as convolution neural network, deep belief networks and recursive neural network, have been successfully used in computer vision, speech recognition, natural language processing, audio identification and bioinformatics, etc. [11].

The convolutional neural network (CNN) is a type of deep learning models in which feature extraction from the input data is done using layers comprised of convolutional operations [12]. The basic structure of CNN, shown in Figure 1, includes input layer, convolutional layer, pooling layer, full connection layer and output layer.

![Figure 1. Structure of convolutional neural network](image)

The convolutional and pooled layers are alternately connected. That is, a convolutional layer is connected to a pooling layer, and which is then connected to a convolutional layer. There can be multiple convolutional layers and pooling layers. The convolutional layer is mainly used to automatically extract the features of the input data. For the feature map obtained by the convolutional layer, the pooling layer usually divides it into several adjacent small regions and further simplifies it to a single value. The process is also called pooling operation. There are many pooling functions, such as maximum pooling, mean pooling, $L_2$ pooling and so on. The output layer is set by the specific application task. If it is a classification, the output layer of CNN is usually a classifier.

In a fully connected network, the neurons of each layer are usually connected to all the neurons of the previous layer. However, in the CNN network, each neuron of the convolutional layer, processed by the kernel function (also called the filter), is only connected to the neurons in a local window of the upper layer. So the connections between adjacent layers are greatly reduced. Moreover, the weights of the traditional neural network are only used once to calculate the output of a layer, so many parameters need to be stored. In the CNN network, the weights between adjacent layers are same, that is, the parameters are shared in the network, and then the number of parameters is greatly reduced too.
3. IPv6 Traffic Classification Model

Network traffic is a kind of sequential data, which can form two-dimensional grid structure data after processing. Therefore, it is very suitable to be classified by the convolutional neural network. In this section, a traffic classification method based on convolutional neural network is proposed for IPv6 traffic data, which need not manual selection of traffic features. Specifically, it includes two key steps: IPv6 traffic data processing and convolutional neural network training.

3.1. IPv6 Traffic Dataset

A reasonable and sufficient traffic dataset is a key factor to verify the performance of the classification algorithm. At present, the commonly used traffic classification datasets include Cambridge University MOORE_DATA2, Brescia University UNIBS_DATA, and private datasets collected by the researchers using data crawler tools. However, most of the datasets are IPv4 traffic, and there are no public pure IPv6 traffic datasets. Then we build a pure IPv6 traffic dataset. The dataset consists of the application label and the traffic data. The traffic data are captured by Wireshark tool in the pure IPv6 campus network of a university, including the operations of communicating in the Google instant messaging platform, browsing websites and videos on YouTube and Tsinghua University, operating on a remote host with SSH client, transferring files with SFTP and so on. The dataset contains 8 pure IPv6 applications, in total 232395 records. For example, a hexadecimal traffic record of the YouTube video application is shown in Figure 2.

![Figure 2. Example of IPv6 Youtube video traffic data](image)

3.2. Data Pre-processing

The first step of the method is the pre-processing of the IPv6 traffic dataset, which are stored in the pcapng files. It deletes irrelevant data fields, and transforms them into two-dimensional matrices to meet the need of the input layer of CNN.

![Figure 3. Data pre-processing](image)
of the packets are different each other. Considering the size of the input of CNN is same, then we need to truncate the packets or fill them with random numbers between 0 and 255. For the long packets, their characteristics have been contained in the front of the packet. And for the short packets, the random filling bytes is not features, and would not affect the accuracy of the classification. Considering both the accuracy and the efficiency of data processing, we set the packet length is 900 bytes to generate a 30*30 byte matrix. Such data pre-processing is simple and effective, which can generate quickly the byte matrix while the characteristics of the data retained. Secondly, some data noise will be filtered in the process of data cleaning, including the packets obviously does not belong to this category (such as DNS, ICMPv6, etc.). At the same time, in order to reduce the influence of fixed IP address on the accuracy of traffic classification, the physical address and logical address of packet are randomized in the pre-processing process. That is, the source and the target MAC, the source and the target IPv6 address are filled with random numbers between 0 and 255. Finally, in order to improve the performance of training, the packets need to be normalized. Because the value of each packet is 0–255, then all packets are divided by 256 to normalize.

3.3. Classification model based on CNN

Different convolutional neural network structures have different traffic classification results. Considering the classification accuracy and the algorithm time complexity, and through many trials, the structure of CNN in the proposed method is shown in Figure 4. It consist of one input layer, two convolutional pooling layers, one full connection layer and one output layer.

![IPv6 traffic classification model based on CNN](image)

Figure 4. IPv6 traffic classification model based on CNN

1) The input of the model is the byte matrix which is pre-processed by the method in Section 3.2.
2) The first convolution layer carries out convolution operation. And it has 8 filters with the window size of [3,3], and the moving step size is 1. The output results are computed by the activation functions of Rectified Linear Unit (ReLU). The ReLU function is defined in Equation (1) ($a(x)$ is the output).

$$a(x) = \max[x, 0]$$  

(1)

3) The pooling function adopted by the first pooling layer is the maximum pooling function, its size is [2,2], and the move step size is 1. The maximum pooling function is defined in Equation (2).

$$\text{max\_pooling} \left[ x_1 \ x_2 \ x_3 \ x_4 \right] = \max \left[ x_1, x_2, x_3, x_4 \right]$$  

(2)

4) The second convolution layer has 16 filters with the size of [3,3] and the moving step size of 1. The output results are also computed by the ReLU function.

5) The parameters of the second pooling layer are as same as the first pooling layer.

6) The full-connection layer transforms the two-dimensional features of input data into one-dimensional vectors, which is mainly used for the following classification. Considering the number and accuracy of parameters, this proposed method adopts one full-connection layer by adjusting parameters through experiments.

7) The output layer uses the $\text{softmax}$ function to realize classification. It has 8 neurons corresponding to 8 different IPv6 traffic applications. Suppose the input is $x$, then the output function $\text{softmax}$ is defined in Equation (3).
8) In the training process, Cross Entropy function (CrossEntropy) is used as the loss function. Then the cost function is defined in Equation (4).

\[ J(w, b) = -\frac{1}{8} \sum_{i=1}^{8} [y_i \ln \hat{y}_i + (1 - y_i) \ln(1 - \hat{y}_i)] + \frac{\lambda}{2} \| w \|^2 \]  

(4)

The first right term is the conventional cross entropy, and the second term is the regular term \( \lambda \), which \( \lambda \) is the regular parameter.

9) The back propagation function use the random gradient descent algorithm to update the weight \( w \) and bias \( b \). Assuming that the learning rate is \( \rho \), the iteration update function is defined in Equation (5).

\[ w = w - \rho \frac{\partial J}{\partial w}, \quad b = b - \rho \frac{\partial J}{\partial b} \]  

(5)

4. Experimental test and result analysis

4.1. Experimental Settings
The experiment use the deep learning framework TensorFlow, and the Python dpkt to collect data. The hardware platform is NVIDIA GPU (Tesla T4). TensorFlow version is 2.0.0 GPU and is drive by CUDA10.0.130 and CUDNN 7.6.5. In the training, the dataset was randomly divided into two parts: 80% as the training set, and the remaining 20% as the test set. The experiment is executed 10 epochs, and the loss function and optimizer are categorical_crossentropy and adam function respectively.

4.2. Performance parameters
In this paper, Precision, Recall and F1_score are used to evaluate the performance of the model [13]. Theirs definitions are in Equation (6), Equation (7) and Equation (8).

\[ \text{Precision} = \frac{TP}{TP + FN} \]  

(6)

\[ \text{Recall} = \frac{TP}{TP + FP} \]  

(7)

\[ F_1 \text{-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \]  

(8)

Where, TP (True Positive) is defined as the percentage of samples in category X that are correctly classified as category X. FN (False Negative) is defined as the percentage of samples in category X that were misclassified into other categories. FP (False Positive) is defined as the percentage of samples in other categories that are incorrectly classified as category X. F1_score is the weighted harmonized average of the precision rate and recall rate, which is used to comprehensively reflect the overall performance index.

4.3. Result analysis
Table 1 presents the experimental results of the method of IPv6 traffic classification based on CNN. As you can see, the values of the Precision, Recall, and F1-score of the classification of the 8 pure IPv6 application categories is high.
Table 1. Results of IPv6 traffic classification

| Application  | Precision | Recall | F1-score | Support |
|--------------|-----------|--------|----------|---------|
| web          | 0.99      | 1      | 1        | 7949    |
| youtube      | 0.97      | 0.98   | 0.98     | 2859    |
| hang_chat    | 0.93      | 0.84   | 0.88     | 599     |
| hang_video   | 1         | 1      | 1        | 1993    |
| scopy        | 0.98      | 1      | 0.99     | 1534    |
| CNN_videl    | 0.99      | 1      | 1        | 5400    |
| stelnet      | 1         | 0.98   | 0.99     | 1941    |
| gmail        | 0.95      | 0.96   | 0.96     | 946     |
| micro avg    | 0.99      | 0.99   | 0.99     | 23221   |
| macro avg    | 0.98      | 0.97   | 0.97     | 23221   |
| weighted avg | 0.99      | 0.99   | 0.99     | 23221   |
| avg samples  | 0.99      | 0.99   | 0.99     | 23221   |

Figure 5 shows the change trend of accuracy rate (epoch = 10) in the training process, and Figure 6 shows the change of loss rate (epoch = 10) in the training process. It can be clearly seen that when the epoch is greater than 7, the accuracy rate is stable at over 99%, while the loss rate is below 0.05.
5. Conclusion
The IPv6 traffic classification model based on CNN constructs the optimal classification model, which considers the aspects of network structure, parameter space and parameter optimization on the basis of traffic data pre-processing. The model autonomic learns data features in the convolutional layer, and solves the problem of feature selection in traditional traffic classification algorithms based on machine learning. Finally, the model was tested through the IPv6 traffic dataset collected from the campus network. The experimental results show that the designed model has a good performance in IPv6 traffic classification accuracy, recall rate and F1_score.

Acknowledgments
This work was supported by the CERNET Innovation Project (NGII20190108).

References
[1] CNGI (China's Next Generation Internet)(2020). http://www.cernet.edu.cn/xxh/ji_shu_ju_le_bu/cernet2_lpv6/.
[2] Wang R, Zhang L, Liu Z. (2018) Sub space streaming classification method for campus IPv6 network traffic data. Journal on Communications, 39(S1):44-52.
[3] Dainotti A, Pescape A, Claffy K.C. (2012) Issues and future directions in traffic classification. IEEE Network, 26(1):35-40.
[4] Finsterbusch M., Richter C., Rocha E., et al. (2014) A survey of payload based traffic classification approaches. IEEE Communications Surveys & Tutorials, 16(2):1135-1156.
[5] Hardeep S. (2015) Performance analysis of unsupervised machine learning techniques for network traffic classification. In: 5th International Conference on Advanced Computing and Communication Technologies. Piscataway: IEEE Press. pp. 401-404.
[6] Chen Z, Liu Z, Peng L, et al. (2017) A novel semi-supervised learning method for Internet application identification. Soft Computing, 21(8):1963-1975.
[7] Fan Z., Liu R. (2017) Investigation of machine learning based network traffic classification. In: the International Symposium on Wireless Communication Systems. Berlin: VDE Verlag GmbH, pp. 1-6.
[8] Wang W, Zhu M, Wang J, et al. (2017) End-to-End Encrypted Traffic Classification with One-dimensional Convolution Neural Networks. In: the 2017 IEEE International Conference on Intelligence and Security Informatics: Security and Big Data. Piscataway: IEEE Press. pp. 43-48.
[9] GUO Y., ZHANG A. (2020) Classification Method of Android Traffic based on Convolutional Neural Network. Communications Technology, 53(02):432-437.
[10] Wang Y, Zhou H, Feng H, et al. (2018) Network traffic classification method basing on CNN. Journal on communications, 39(01):14-23.
[11] Goodfellow I, Bengio Y, Courville A. (2016) Deep Learning. Cambridge: MIT Press, 1-26.
[12] Krizhevsky A, Sutskever I, Hinton G E. (2017) ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60(6):84-90.
[13] Kim H, Claffy K C, Fomenkov M, et al. (2008) Internet traffic classification demystified: myths, caveats, and the best practices. In: the 4th International Conference on Emerging Networking Experiments and Technologies. New York: ACM Press, pp. 9-12.