Classification and identification of unknown network protocols based on CNN and T-SNE

Jingliang Xue1,*, Yingchun Chen1, Ou Li1, Fei Li2

1School of Information Engineering, PLA Strategic Support Force Information Engineering University, Zhengzhou, China
2School of Foreign Languages, PLA Strategic Support Force Information Engineering University, Zhengzhou, China
*13340247073@163.com

Abstract—With the continuous development of users’ demands and network technology, more and more new network protocols emerge, which poses great challenges to network protocol classification and identification. An artificial intelligence method was used to explore autonomous classification and identification of unknown network protocols in this paper in order to reduce the time and labor cost of network protocol classification and identification. In this paper, firstly, the network traffic was converted into grayscale images, and through transfer learning, the Convolutional Neural Networks (CNN) pre-trained model was used to extract the protocol features, so as to reduce the time and the amount of labeled data needed for the artificial neural network training. Finally, with the improved unsupervised hybrid clustering algorithm based on T-SNE and K-means, the types and number of protocols were autonomously identified and the network traffic was classified simultaneously. In this way, we can identify unknown protocols without prior knowledge and the protocol identification adaptability for big data was also greatly improved. Experimental results show this method has high accuracy and robustness in identifying unknown network protocols.

1. INTRODUCTION
With the increasing scale of network communication and the constant change of people’s needs, more encrypted traffic and private protocols appear on the Internet. The classification and identification of unknown network protocols can provide support for further protocol reverse parsing, and therefore more accurate protocol detection through clustering analysis[1]. Research on classification and identification technology of unknown network protocols can effectively provide technical support for detecting illegal intrusion, monitoring the traffic flow, analyzing user behavior and eventually ensuring network security.

Protocol identification can be achieved through many ways. The traditional method uses fixed port numbers, but such method can be easily cheated by changing the port number in the system [2]. DPI (Deep Packet Inspection) is the most commonly used protocol identification technology at present. It needs to conduct further in-depth inspection on the header, payloads and other information of data packets. However, it cannot identify unknown protocol types, and its feature database may cause heavy resource consumption [3]. The method based on association rule mining for unknown protocol identification has certain limitations. For example, in the case of real-time large-scale network protocol analysis, the computational complexity is enormous [4]. Machine learning methods have a powerful adaptive and learning capability, and have developed rapidly in the field of protocol analysis. Generally speaking,
machine learning is mainly divided into unsupervised learning and supervised learning. Unsupervised learning methods are often used to identify unknown protocols and can mine data features without category information. Hong et al. [1] proposed an application layer protocol classification and identification method which combines the traditional DPI technology and clustering methods to adapt to the number of target clusters, and can efficiently classify and identify unknown application layer protocols. Peng et al. [5] used mathematical statistics to calculate the K value and the cluster initial center of the K-means clustering algorithm and realized data clustering. Zhang et al. [6] combined the traditional AGNES hierarchical clustering algorithm with the features of the bitstream data frames, and proposed a classification method for protocols with unknown bitstreams. This method can automatically identify the number of clusters and classify unknown bitstream data frames. However, most protocol identification methods based on traditional machine learning require manual feature selection as input in order to further classify and identify protocols.

Supervised learning is a method that trains models to predict identification results. Deep learning is a typical supervised learning method, and can convert data into data that can be learned by machines. It autonomously transforms low-level features into complex high-level features for representing the attributes of input images, in order to learn the inherent rules and the representation levels of sample data. This end-to-end learning method is free from the complex steps of extracting features in advance and increases the automation level of the protocol classification and identification. In real-time analysis of online network traffic and big data volume analysis, such as image and video classification and identification analysis, this method has achieved good results. Wang et al. [7] first proposed the idea of treating the bit data of traffic as pixels of an image and applied deep learning to traffic classification and identification. Based on the similarity between network traffic and images, Zhang et al. [8] directly used network traffic data as input of CNN to train the classification and identification capability for the model. Wang et al. [9] was the first to realize the classification of malware by using the characterization learning method of raw data, and improved the accuracy of the classification and identification. Li et al. [10] proposed a Byte Segment Neural Network (BSNN). This Neural Network does not require a priori knowledge and can handle both connection-oriented and connectionless protocols simultaneously. Deep learning has achieved success in protocol classification and identification. However, they depend too much on labeled data, and there is a lack of recognized datasets for protocol classification and identification. Most researchers adopt raw traffic data captured under their respective experiment network conditions and the data are always labeled by category through manual methods or DPI tools, which has low accuracy and complicated steps [11]. In addition, how to use deep learning methods to distinguish between known and unknown network protocols in data traffic and analyze unknown protocols is still a problem in research on network protocol classification and identification.

This paper proposed a new classification and identification method for unknown network protocols, and this method has the advantages of both deep learning and unsupervised learning. It does not rely too much on data to train the model, and can directly use CNN to obtain the features of unknown protocols. In this paper, first, CNN with pre-trained model weight was used to automatically extract the features of unknown network protocols. Then, through the improved dimension reduction algorithm of T-SNE, the dimensions of the features were intelligently reduced and the number of unknown protocols is identified. Finally, using the distance selection feature of the K-means algorithm, we directly realized the unknown protocol classification of the traffic data.

2. METHODS

2.1 Data pre-processing
In order to facilitate the analysis and processing of the unknown network protocols in the later stage, the traffic data captured from the network need to be pre-processed through three steps: payload extraction, data conversion and image generation.

**Step 1** (payload extraction): extract the payload part of the traffic information in the network traffic packet to facilitate further analysis of the traffic data, such as using Scapy to process the .pcap file.
Step 2 (data conversion): uniformly convert the hexadecimal data of the payload part into binary bitstreams to facilitate the subsequent generation of the grayscale images.

Step 3 (image generation): convert binary bitstreams into grayscale images. The binary value 1 corresponds to gray value 256, and 0 corresponds to 0. As the lengths of the binary bitstream vary, it is not conducive to generating regular square images that can be recognized by CNN. Here, it is stipulated that the binary bitstream with insufficient length will be supplemented with 0 at the end. The specific rules are as follows, if $(n - 1)^2 < l < n^2$, $n^2 - l$ 0s must be added at the end of the binary bitstream. $n$ represents the pixel value of the edge, and $l$ represents the length of the binary bitstream. Finally, the converted gray values were stored in the matrix in the form of $n \times n$ in order, and saved in an image format.

2.2 Intelligent feature extraction

1) Feature extraction structure

CNN is a very effective image identification algorithm in deep learning. It is mainly used to identify the graphics with distortion invariance, such as scaling, displacement and others. CNN optimizes the loss function through iterative training, avoids explicit feature extraction, and can learn features implicitly from the training data.

![Figure 1. Example of a CNN structure](image)

The basic CNN structure consists of two parts, as shown in Fig. 1. One is the feature extraction part, which is made up of alternate convolution layers and pooling layers. The convolution layer convolved the images and the filters to extract the local features of the image. The pooling layer shrinks the input images, reduces pixel information and retains important information. The second part is the feature mapping part, which can also be called the classification and identification part and includes fully connected layers. The first fully connected layer maps the latent feature space processed by the previous layer to a distributed feature representation. The last fully connected layer is a classifier that maps distributed features to the label space to classify the input image.

In this paper, only the feature extraction part of the CNN was used. The output of the feature extraction part was the effective features of the input image obtained autonomously by the CNN.

2) Transfer learning

Transfer learning refers to the transfer of labeled data or knowledge structures from related fields for completion or improvement of the learning effect of the target field or task. Transfer learning is based on the assumption that the processing mechanisms of neural networks are similar to those of human brains which is continuous and iterative, and neural networks can also identify new things based on existing knowledge. In deep learning, transfer learning trains the CNN model to learn network parameters on a certain large dataset, and then applies it to another dataset. The advantage of transfer learning is that the pre-trained model can classify completely different datasets, and share the pre-trained weights of the deep neural network structure and apply it to our own dataset. This method significantly reduces the time and
the labeled data required for training. Transfer learning can be roughly divided into: instance-based deep transfer learning, mapping-based deep transfer learning, network-based deep transfer learning, and others [12]. Keras pre-trained models LeNet, AlexNet, VGG, Inception and ResNet deliver good performance in network-based deep transfer learning [13]. Keras pre-trained models usually refer to convolutional neural networks trained on ImageNet, which are generally used for the architecture of vision-related tasks. The ImageNet dataset used for training contains approximately 1 million images, which can be divided into 1,000 categories [14].

2.3 Dimension reduction identification

After autonomous feature extraction, the data to be processed should be clustered to achieve the classification and identification of the unknown protocols. However, as the number of unknown protocols is not certain and the feature vectors output by the CNN tend to have high dimensions, the use of traditional clustering algorithms is limited. To solve these problems, a hybrid dimension reduction clustering algorithm based on the combination of T-Distributed Stochastic Neighbour Embedding (T-SNE) [15] and the K-means algorithm was proposed in this paper. With the hybrid dimension reduction clustering algorithm, the problems of high data feature dimensions and the K-means algorithm’s inability to classify and identify protocols without knowing the cluster number are solved.

T-SNE is a nonlinear dimension reduction algorithm. It has the ability to project the high-dimensional data into a low-dimensional space for visualization while maintaining the local structure. The problem of crowding and difficulty of optimization in the traditional SNE algorithm is solved by using T-distribution which pays more attention to long-tailed distribution in low-dimensional space and increases the distance between different clusters.

With \( X = \{x_1, x_2, \ldots, x_n\} \) as the input space, \( Y = \{y_1, y_2, \ldots, y_n\} \) is the space after dimension reduction. T-SNE first calculates the conditional probability \( p_{ji|j} \) according to the Euclidean distance between data points \( x_i \) and \( x_j \). \( p_{ji|j} \) is expressed in (1):

\[
p_{ji|j} = \frac{\exp(-\|x_i-x_j\|^2/2\sigma_i^2)}{\sum_{k=1}^{n} \exp(-\|x_i-x_k\|^2/2\sigma_i^2)},
\]

where \( \sigma_i \) has different values for different point \( x_i \), and the Gaussian mean square deviation centering on data point \( x_i \) is usually used as its value.

T-SNE minimizes the KL divergence by optimizing the difference between joint probability distribution \( P \) in the high-dimensional space and joint probability distribution \( Q \) in the low-dimensional space, The function can be defined as:

\[
C = KL(P||Q) = \sum_{i} \sum_{j} p_{ij} \log \frac{p_{ij}}{q_{ij}},
\]

where \( p_{ij} \) and \( q_{ij} \) are the joint probabilities of high-dimensional space and low-dimensional space, respectively. The calculation formulas can be defined as:

\[
p_{ij} = \frac{p_{ji}+p_{ij}}{2n},
\]

\[
q_{ij} = \frac{(1+\|y_i-y_j\|^2)^{-1}}{\sum_{k=1}^{n} (1+\|y_k-y_j\|^2)^{-1}}.
\]

T-SNE uses the gradient descent method to solve the optimization objective problem, so the optimized gradient can be obtained, as shown in (4):

\[
\frac{\delta C}{\delta y_j} = 4 \sum_{i} (p_{ij} - q_{ij}) (y_i - y_j) (1 + \|y_i - y_j\|^2)^{-1},
\]

The iterative formula of the output vector is shown in (5):
\[ y^{(t)} = y^{(t-1)} + \eta \frac{\delta C}{\delta y} + \alpha(t)(y^{(t-1)} - y^{(t-2)}), \]  

(5)

where \( y^{(t)} \) is the solution to the t-time iterations, \( \alpha(t) \) is the momentum of the t-time iterations, and \( \eta \) is the learning rate.

Figure 2. Flowchart of the T-SNE algorithm

The overall flowchart of the T-SNE algorithm is shown in Fig. 2. Considering that T-SNE involves many calculations, such as conditional probability and gradient descent, the complexity of time and space is of the quadratic level. It consumes a lot of resources when the data dimensions are very high. PCA is a linear calculation with fast calculation speed. In this paper, we tried to combine the nonlinear dimension reduction of T-SNE with the linear dimension reduction of PCA to reduce the computation amount and running time while ensuring certain stability of the data's internal structure. Through the above-mentioned dimension reduction algorithm, we significantly reduced the dimension of features and determined the unknown number of feature clusters in the visualization analysis. It laid a foundation for the next step of traffic classification based on K-means [16]. K-means has the advantages of fast convergence speed, a better clustering effect, and relatively strong interpretability of the model. The K-means algorithm determines the centroid of each cluster through iterative training. Once the iteration is over, the centroid of each cluster is also determined, and the data points that have participated in the training are close to their nearest centroid. Finally, the traffic data corresponding to the data points are classified by calculating the distance between the data points and the centroid of each cluster.

The overall flowchart of the dimension reduction identification for high-dimensional feature is shown in Fig. 3.

Figure 3. Overall flowchart of the dimension reduction identification for high-dimensional feature

In Fig. 3, the dimensions of the high-dimensional feature vectors were first reduced to 50 dimensions through PCA, and then to 2 dimensions through T-SNE. Finally, the K-means algorithm was used to realize the classification and identification of the traffic data.
3. EXPERIMENTS

The experimental dataset in this paper was the actual network traffic data captured by Wireshark, we selected the unencrypted traffic data for testing, including 12 protocol types such as common application layer protocols of HTTP, DNS, SMTP, and FTP, and private application protocols of OICQ, WOW, and others. The selected traffic data were saved in the .pcap format, and the pre-processed traffic images were saved in the .jpg format.

In order to better analyze the classification and identification performance of the algorithm proposed in this paper, the following four performance indicators were used in the experimental test: accuracy, precision, recall and F1 score. Among them, the F1 score is the main indicator, which is the weighted average of precision and recall indicators. An F1 score of 1 indicates that the algorithm performance in the test is the best, while 0 is the worst.

3.1 Dataset pre-processing

According to the data pre-processing process, the payload in the traffic protocol data packet was first extracted. The DNS protocol data information captured in Wireshark is shown in Fig. 4. and Fig. 5 shows the complete hexadecimal content of a single DNS data. The payload of the DNS data after extraction is shown in the highlighted part of Fig. 6. The binary form of the extracted payload part after data conversion is shown in Fig. 7. For the DNS protocol data, the generated image after data pre-processing is shown in Fig. 8.
Figure 9. Feature images of 12 different protocol types

We performed data pre-processing for the captured network traffic data of the 12 different protocol types, and the grayscale images obtained are shown in Fig. 9. There are local texture features in the images, which reflect the protocol characteristics to a certain extent. Image texture features were extracted using the CNN and characterized the protocol format information to a certain extent.

3.2 Comparison test of pre-trained models

In order to analyze the influence of different CNN pre-trained models on the classification and identification results of unknown protocol data, six pre-trained models were selected for a comparison test in this paper. The model parameters are shown in Table I. The CNN models were implemented using Keras and Tensorflow backends [13], and the performance indicators were calculated using scikit-learn in Python.

| Model      | Size   | Top-1 Accuracy | Top-5 Accuracy | Parameters | Depth |
|------------|--------|----------------|----------------|------------|-------|
| ResNet-50  | 99 MB  | 0.749          | 0.921          | 25,636,712 | 168   |
| VGG16      | 528 MB | 0.713          | 0.901          | 138,357,544| 23    |
| VGG19      | 549 MB | 0.713          | 0.9            | 143,667,240| 26    |
| Inception V3 | 92 MB  | 0.779          | 0.937          | 23,851,784 | 159   |
| Xception   | 88 MB  | 0.79           | 0.945          | 22,910,480 | 126   |
| MobileNet  | 16 MB  | 0.704          | 0.895          | 4,253,864  | 88    |

* Taken from Home - Keras Documentation (2020)

All pre-trained models were tested with the same experimental parameters, including the number of iterations. We randomly selected 150 traffic images of each of the three protocols of DNS, Facetime, and HTTP as the test set. Considering that the different payload contents cause the image pixels to be non-uniform, we uniformly reshaped the images to a size of $128 \times 128$, and the PCA+T-SNE+K-means dimension reduction clustering algorithm was used. The average classification and identification results of the three protocols are shown in Table II.

| Pre-trained Model | Accuracy | F1 Score | Precision | Recall |
|-------------------|----------|----------|-----------|--------|
| ResNet-50         | 0.8978   | 0.8967   | 0.8966    | 0.8978 |
| MobileNet         | 0.8956   | 0.8948   | 0.8970    | 0.8956 |
| Xception          | 0.8222   | 0.8160   | 0.8291    | 0.8222 |
| Inception V3      | 0.8067   | 0.8080   | 0.8198    | 0.8067 |
| VGG19             | 0.7600   | 0.7579   | 0.7715    | 0.7600 |
| VGG16             | 0.7578   | 0.7587   | 0.7599    | 0.7578 |
Table II ranks the different pre-trained models from top to bottom according to the accuracy. The ResNet-50 model obtained the best result in terms of both accuracy and F1 score.

For DNS, FaceTime and HTTP protocols, we used the ResNet-50 pre-trained model to analyze each protocol in more details. The clustering confusion matrix for the three protocols is shown in Fig. 10. Coordinate labels 1, 2 and 3 correspond to three types of protocols: DNS, HTTP and FaceTime. The sum of each column is the predicted number of the protocol category, the sum of each row is the actual number of the protocol category. Among them, the classification and identification result of DNS is the best, and a small number of HTTP protocol instances are confused with FaceTime protocols. The classification and recognition results of each protocol are shown in Table III. The F1 scores of the three protocol types are all higher than 84%, with that of DNS being the highest, reaching 97.09%. The overall accuracy is 89.78% and the average F1 score is 89.78%.

![Confusion matrix analysis of three protocols on ResNet-50](image)

**TABLE III. Classification and identification results of the ResNet-50 pre-trained model**

| Protocol Category | Precision | Recall | F1 Score | Support |
|-------------------|-----------|--------|----------|---------|
| DNS               | 0.9434    | 1      | 0.9709   | 150     |
| HTTP              | 0.8514    | 0.84   | 0.8456   | 150     |
| FaceTime          | 0.8951    | 0.8533 | 0.8737   | 150     |
| Accuracy          |           |        |          |         |
| Macro average     | 0.8966    | 0.8978 | 0.8978   | 450     |
| Weighted average  | 0.8966    | 0.8978 | 0.8978   | 450     |

3.3 Comparison experiment of dimension reduction algorithms
The experiments in this section mainly compares the influence of the three dimension reduction algorithms, T-SNE, PCA+T-SNE, and PCA, on the classification and identification of unknown network protocols. Besides the dimension reduction algorithm itself, the perplexity setting of T-SNE and the pre-reduction dimensions of PCA in PCA+T-SNE will have some influence on the experimental results. Through experimental analysis, it was found out that the classification and identification accuracy of the T-SNE algorithm was stable when the perplexity was changed, while the changes of the perplexity and the PCA pre-reduction dimension in the T-SNE+PCA algorithm affected the accuracy. When the perplexity was set to 50 and the PCA pre-reduction dimensions were set to 50, the optimal classification and identification results were obtained. Based on the above-mentioned parameters, the ResNet-50 pre-trained model was selected to classify and identify the three protocols of DNS, FaceTime, and HTTP. The classification and identification results of unknown network protocols under three different
dimension reduction algorithms are shown in Table IV, and the results are the average classification and identification results of the three protocols.

### TABLE IV. Classification and identification results of three dimension reduction algorithms

| Algorithm   | Accuracy | F1 Score | Precision | Recall  |
|-------------|----------|----------|-----------|---------|
| T-SNE       | 0.8978   | 0.8967   | 0.8966    | 0.8978  |
| PCA+T-SNE   | 0.9      | 0.8991   | 0.9       |         |
| PCA         | 0.8867   | 0.8862   | 0.8859    | 0.8867  |

It can be seen from Table IV that T-SNE has better performances than PCA. The main reason is that PCA is a linear dimension reduction algorithm, which has difficulty in explaining the complex polynomial relationship between features, while T-SNE finds out the structural relationship in data by calculating the random probability distribution on the neighborhood graph. There was not much difference between the results of T-SNE and PCA+T-SNE, but the integration of PCA and T-SNE can reduce the calculation amount and time while ensuring the accuracy of the result. Therefore, the combined dimension reduction algorithm of PCA and T-SNE was adopted in this paper.

Fig. 11 shows the results of the PCA+T-SNE algorithm after reduction and visualization of high-dimensional protocol features, with red representing DNS, blue HTTP, and green FaceTime.

![Figure 11. Result of PCA+T-SNE dimension reduction](image-url)

### 3.4 Robustness test

Aiming at the problems of data errors and losses during the actual network data transmission, the ResNet-50 pre-trained model and the PCA+T-SNE dimension reduction algorithm were used in this experiment to test the robustness of the classification and identification method. This test mainly included two indicators: packet loss rate and bit error rate. Table V and Table VI show the average classification and identification results of DNS, FaceTime and HTTP data under the interference of packet loss rate and bit error rate, respectively.

### Table V. Effect of packet loss rate on the classification and identification results

| Packet Loss Rate | Accuracy | F1 Score | Precision | Recall  |
|------------------|----------|----------|-----------|---------|
| 0.1%             | 0.8956   | 0.8945   | 0.8942    | 0.8956  |
| 1.0%             | 0.8978   | 0.8967   | 0.8966    | 0.8978  |
| 5.0%             | 0.8867   | 0.8862   | 0.8859    | 0.8867  |
| 10.0%            | 0.8978   | 0.8967   | 0.8968    | 0.8978  |
TABLE VI. EFFECT OF BIT ERROR RATE ON THE CLASSIFICATION AND IDENTIFICATION RESULTS

| Bit Error Rate | Accuracy | F1 Score | Precision | Recall |
|----------------|----------|----------|-----------|--------|
| 0.1%           | 0.8978   | 0.8969   | 0.8968    | 0.8978 |
| 1.0%           | 0.8956   | 0.8946   | 0.8945    | 0.8956 |
| 5.0%           | 0.8933   | 0.8923   | 0.8921    | 0.8933 |
| 10.0%          | 0.8733   | 0.8722   | 0.8733    | 0.8733 |

It can be seen from the two tables that the protocol classification and identification accuracy did not deteriorate significantly when the packet loss rate and the bit error rate changed from 0.1% to 10%. This shows that the algorithm proposed in this paper, which converts the protocol data into grayscale images as input, and uses intelligent algorithms for classification and identification, has good robustness. The experimental results verify the effectiveness of this algorithm.

4. CONCLUSIONS

A classification and identification method for unknown network protocols based on CNN and T-SNE was proposed in this paper. Through this method, first, the protocol data payload information from the network traffic was extracted. Then, the payload information was converted into grayscale images, and the CNN pre-trained model was used to extract features as the basis for protocol classification and identification. Finally, dimension reduction clustering algorithms based on T-SNE and K-means were adopted to intelligently cluster the feature vectors to efficiently and accurately realize the classification and identification of unknown network protocols. This method made full use of the advantage of CNN’s end-to-end learning. On the basis of ensuring the classification and identification accuracy, it avoided the complex steps of manually extracting features and reduced the training time of the intelligent algorithm as well as the amount of labeled data required.

This article is a preliminary exploration of deep metric learning in the identification of unknown protocols, the protocol feature embeddings in the traffic information are extracted through the neural network, and the protocol clustering and recognition can be realized through these standardized feature embeddings. It turns out that the features extracted by the neural network are indeed can represent part of the information of the protocol and has certain validity in the identification of unknown protocols. In the future, we hope to combine the LMNN idea to optimize the CNN feature output process, increase the feature similarity of the same protocol data and widen the differences between different protocol data to improve the model's representation capability. We will also do further research on encrypted traffic, and try to use neural networks to find the potential characteristics of encrypted data.

ACKNOWLEDGMENT

This research was funded by the National Natural Science Foundation of China (61601516).

REFERENCES

[1] Hong Z, Gong Q, Feng W, Li Y. Unknown Application Layer Protocol Identification Based on Adaptive Clustering. Computer Engineering and Applications. 2020, 56(05): 109-117.

[2] Moore A W, Zuev D. Internet traffic classification using bayesian analysis techniques[C]/Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems. 2005: 50-60.

[3] Guo L. Research on Multi-Business Identification Technology Oriented High-Speed Network Management and Control. Doctor, The PLA Information Engineering University, Zhengzhou, Henan, China, 2012.

[4] Lou S. Research on Parallel FP-Growth Association Rule. Master, University of Electronic Science and Technology of China, Chengdu, Sichuan, China, 2016.
[5] Peng D, Xiang L, Li S, Yang C, Qiu Y. Classification of intelligent home protocol under multi-protocols. Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition), 2018, 30(03): 321-328.

[6] Zhang F, Zhou H, Zhang J, Liu Y, Zhang C. A protocol classification algorithm based on improved AGNES. Computer Engineering and Science, 2017, 39(04): 796-803.

[7] Wang Z. The applications of deep learning on traffic identification[J]. BlackHat USA, 2015, 24(11): 1-10.

[8] Zhang, L.; Liao, P.; Zhao, J.; Guo, L. A Method of Unknown Protocol Identification Based on Convolution Neural Network. Microelectronics & Computer, 2018, 35(07): 106-108.

[9] Wang W, Zhu M, Zeng X, et al. Malware traffic classification using convolutional neural network for representation learning[C]//2017 International Conference on Information Networking (ICOIN). IEEE, 2017: 712-717.

[10] Li R, Xiao X, Ni S, et al. Byte segment neural network for network traffic classification[C]//2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS). IEEE, 2018: 1-10.

[11] Feng W, Hong Z, Wu L, Fu M. Review of network protocol identification techniques. Computer Applications. 2019, 39: 3604-3614.

[12] Tan C, Sun F, Kong T, et al. A survey on deep transfer learning[C]//International Conference on Artificial Neural Networks. Springer, Cham, 2018: 270-279.

[13] Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Advances in neural information processing systems. pp. 3320–3328 (2014).

[14] Keras – Home. Keras Documentation. Available online: https://keras.io/ (accessed on 8 February 2020).

[15] Maaten L, Hinton G. Visualizing data using t-SNE[J]. Journal of machine learning research, 2008, 9(Nov): 2579-2605.

[16] MacQueen J. Some methods for classification and analysis of multivariate observations[C]//Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. 1967, 1(14): 281-297.