



Deep learning-based real-time VPN encrypted traffic identification methods

Lulu Guo¹ · Qianqiong Wu^{2,3} · Shengli Liu¹ · Ming Duan¹ · Huijie Li¹ · Jianwen Sun¹

Received: 30 August 2019 / Accepted: 15 November 2019 / Published online: 2 December 2019
© Springer-Verlag GmbH Germany, part of Springer Nature 2019

Abstract

With the widespread application of virtual private network (VPN) technology, real-time VPN traffic identification has become an increasingly important task in network management and security maintenance. Since traditional encrypted traffic identification technology is not effective in feature extraction and selection, this paper proposes two deep learning-based models to classify the traffic into VPN and non-VPN traffic, identify VPN traffic generated by six different applications much further. Our models utilize convolutional auto-encoding (CAE) and convolutional neural network (CNN), respectively, preprocessing the traffic samples into session pictures, to accomplish the experiment objectives. The CAE-based method, utilizing the unsupervised nature of CAE to extract the hidden layer features, can automatically learn the nonlinear relationship between original input and expected output. The CNN-based method performs well in extracting two-dimensional local features of images. Experimental results show that our models perform better than traditional identification methods. In the two-category identification, the best result comes from the CAE-based model; the overall identification accuracy rate is 98.77%. Among the six-category identification, the best result comes from CNN-based model; the overall identification accuracy rate is 92.92%.

Keywords CAE · CNN · Real time · VPN · Identification

1 Introduction

Nowadays, virtual private network (VPN) technology is widely used in network communication to meet different security requirements as people pay increasing attention to communication security. The rapid growth of high-throughput business requires higher quality of service (QoS) and better network resource management, and the IP traffic identification plays an important role in improving the level of QoS and network resource management. The encrypted traffic identification, especially real-time VPN encrypted traffic identification, is an important and challenging task in IP traffic identification, so it is very practical to research real-time VPN encrypted traffic identification.

Traditional VPN identification technology is mainly based on port filtering, deep packet inspection (DPI), machine learning, and statistical methods [1]. DPI technology maps key features such as time domain features and handshake protocol fields generated during the interaction process to distinguish VPN encrypted traffic [2, 3]. Machine learning methods generally use different machine learning algorithms to perform training with time-related features extracted from the encrypted traffic [4–6]. There are also some methods based on entropy, fingerprint, and host behavior. The extraction of features and key protocol fields in the above methods is often complicated, costly, and time consuming. The encrypted traffic cannot be recognized if key features are modified or extracted with errors. Therefore, the deep learning-based method for traffic identification came into being. Zhanyi Wang first applied deep learning to traffic identification, mapped TCP traffic data into pictures or documents, and conducted feature training through ANN and SAE deep learning algorithms [7]. It has achieved certain effects in the field of open protocol classification and unknown protocol identification, but not involved in encrypted traffic. Lotfollahi proposed integrating feature extraction and classification into a system

✉ Lulu Guo
mdscience@sina.com

¹ Cyberspace Security Institute, Information Engineering University, Zhengzhou, China

² Institute of Electrical Engineering and Automation, Luoyang Institute of Science and Technology, Luoyang, China

³ Tampere University of Applied Sciences, Pirkanmaa, Finland

embed in stack self-encoding and convolutional neural network algorithms to differentiate VPN encrypted traffic [8]. However, the discussion of data preprocessing and model processing flow is not clear enough. Wei Wang proposed an end-to-end encrypted traffic identification method based on one-dimensional convolutional neural network. The data preprocessing selects the first 784 bytes of each session, which is more suitable for the one-dimensional sequence characteristics of network traffic than the commonly used two-dimensional convolutional neural network [9]. However, data preprocessing tends to lose important feature information. Ding Li proposed the mobile phone HTTP (Hyper Text Transfer Protocol) application traffic identification method based on convolutional auto-encoding technology [10], encrypted traffic not involved. Xuejiao Chen proposed convolutional neural technology [11], Pan Wang proposed stack-based auto-encoding technology [12]; they conducted the VPN encrypted traffic identification but only with a part of the VPN data set, which is not convenient for comparison testing.

According to current research, this paper, converting the traffic into a special format image, proposes the application of convolutional auto-encoder (CAE) and convolutional neural network (CNN) technology for identifying VPN encrypted traffic. As shown in Fig. 1, the two methods will both distinguish VPN traffic from Non-VPN traffic, which further distinguish six different application types in VPN traffic. The comparison of identification effect between the two methods will be analyzed at the end.

2 Background

2.1 VPN encrypted traffic

VPN refers to a dedicated communication network established in a public network using cryptography and access control technology. The connection between any two nodes does not have the end-to-end physical link required by the

traditional private network, but is dynamically composed of certain resources of the public network. The virtual private network is transparent to the client, and the user seems to use a dedicated line to communicate. VPN is divided into SSL VPN, IPSec VPN, PPTP VPN, and L2TP VPN according to the different tunnel security protocol. SSL VPN, and IPSec VPN are two most popular VPN technologies among all kinds of technologies. This paper focuses on the identification of SSL VPN traffic generated by OpenVPN clients.

2.2 Data set description

To facilitate research and comparison, we use the public VPN–Non-VPN data set (ISCXVPN2016) [4]. The data set was released for research by the ISCX Research Center at the University of New Brunswick (UNB). The research center uses OpenVPN to establish connection with an external VPN service provider and uses TCPDump and Wireshark software to capture traffic. As shown in Table 1, a total of seven categories of application traffic are generated, including web, email, chat, streaming, file transfer, IP voice, and peer-to-peer. These files are divided into two categories, which are packets captured through a VPN session (where VPN traffic does not include the type of web traffic [9]) and Non-VPN session.

2.3 Network traffic description

Network traffic generally has five forms of representation, namely TCP connection, flow, session, service, and host. Flow refers to all packets with the same five tuple (source IP, source port, destination IP, destination port, and transport layer protocol). Session refers to a packet consisting of two-way streams, which means that the source and destination addresses can be exchanged. TCP and UDP sessions, including the same five tuple, are defined as bidirectional streams. Streams and sessions are the two most common forms of expressing network traffic. This paper transforms the public data set into a more conceivable session form for further research on the basis of previous studies [9].

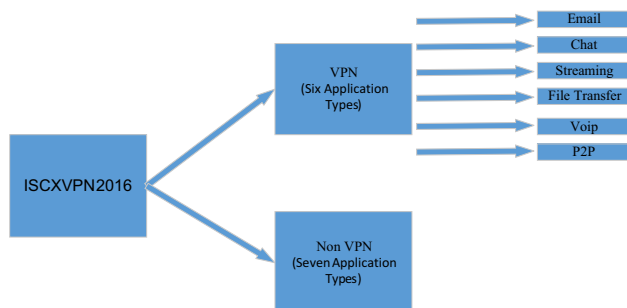


Fig. 1 The classification of real-time VPN encrypted traffic identification

Table 1 ISCX VPN 2016 data set content

Traffic category	Content
Chat	ICQ/AIM/Skype/Facebook/hangouts
Email	SMTPS/POP3S/IMAPS
File Transfer	Skype file/FTPS/SFTP/SCP
P2P	uTorrent/Bittorrent
Stream	Vimeo/YouTube/Netflix/Spotify
VoIP	Voip buster/Facebook audio/Skype/hangouts audio
Web	Firefox/Chrome

2.4 Model

Auto-encoding (AE) was proposed by Kingma and Rezende et al. in 2013 as a generative model. First, AE encodes the input into the low-dimensional potential space, then decodes it back, and then tries to make the input be equal to the output through learning the function in equation [1], thereby finds the hidden structure of the original data. As shown in equation [1], set the original data to \hat{x} , Output Data \hat{x} , the mapping function is f_{enc} and f_{dec} . The connection weights of the input layer, the intermediate hidden layer, and the output layer are continuously adjusted through training:

$$f_{\text{dec}}(f_{\text{enc}}(x)) = \hat{x} \approx x. \quad (1)$$

Convolution auto-encoder (CAE) introduces a convolution operation on the basis of the auto-encoding. The convolution operation preserves the local correlation of the input and can extract the two-dimensional local features of the original input, thereby realizing the global sharing of weights.

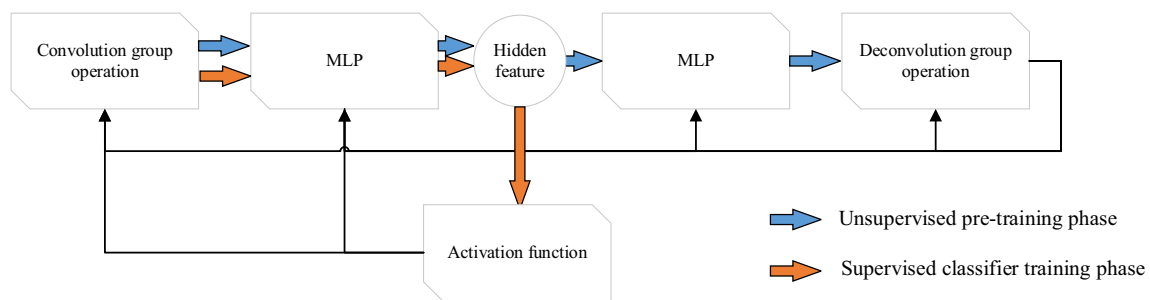
The CAE-based encrypted traffic identification model consists of two phases, as shown in Fig. 2a. The first phase is the unsupervised CAE pretraining process, called unsupervised pretraining process. The second phase is to connect the trained CAE with the classification output, called the supervised classifier training process.

Convolutional neural network (CNN) is one kind of feedback forward neural network consisting of one or more convolutional layers and fully connected layers. It also includes a pooling layer and a padding layer, as shown in Fig. 2b.

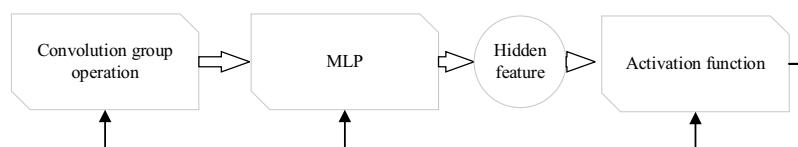
This structure enables CNN to take advantage of the two-dimensional structure of the input data to perform well in image processing.

As shown in Fig. 2, the two models consist of convolution group operations, deconvolution group operations, and full connection operations, where each convolution group includes padding, convolution, and pooling operations. Each deconvolution group contains cropping, deconvolution, and anti-pooling operations. The input size in each layer of the model is set to (H, W) . The process of each specific operation is as follows.

- ① Padding: The padding layer is used to perform a zero-padding operation on the original input to enlarge the size of the image. Assuming that the parameter is (h, w) , respectively, we perform symmetrical zero-padding operation on h lines and w columns in the vertical and horizontal directions of the input; the output size is $(H + 2h, W + 2w)$.
- ② Cropping: The cropping layer is the inverse of the padding layer, mainly used for cropping appropriately the edge of the input picture. Assuming that the parameter is (h, w) , we perform symmetric cropping operation on h lines and w columns in the vertical and horizontal directions of the input. The output size is $(H - 2h, W - 2w)$.
- ③ Convolution: The convolution layer converts spatial position information of the input image into a high-order feature representation [13]. Assuming that input data are x , the No. k convolution kernel is (k_x, k_y) , B is the offset, W^k represents the weight matrix of the feature map of convolution kernel, and $*$ represents the two-



(a) CAE-based identification model



(b) CNN-based identification model

Fig. 2 Two identification models of deep learning

dimensional convolution operation. After the first k convolution kernel operations, the output is as shown in the following formula:

$$x_k = f(x * W^k + B). \quad (2)$$

And the output size is $(H - k_x + 1, W - k_y + 1)$. Where f is the activation function and the rectified linear unit (Relu) is used to accelerate the convergence process.

- ④ Deconvolution: The deconvolution operation can be seen as the reverse of the convolution operation. Assuming that (k_x, k_y) is the No. k deconvolution kernel, x_k is the input, $\overline{W^k}$ is the transposition of the No. k feature map weight matrix of the deconvolution kernel, \hat{B} represents the offset corresponding to the deconvolution core, and the reconstructed output of the No. k deconvolution kernel after decoding is as shown in the following formula:

$$\hat{x} = f(x_k * \overline{W^k} + \hat{B}) \quad (3)$$

And the output size is $(H + k_x - 1, W + k_y - 1)$, and we also choose the rectified linear unit (Relu) as the activation function f .

- ⑤ Pooling: The pooling operation, also called subsampling operation, is used to reduce redundant information in the feature map and extract high-order features. Assuming that (c_x, c_y) is parameter of pooling kernel, the output size is $\left(\frac{H}{c_x}, \frac{W}{c_y}\right)$.
- ⑥ Anti-pooling: The anti-pooling operation, also called the upsampling operation, the inverse process of the pooling layer, should restore the data before sampling as much as possible as the pooling process is a process of data loss. We choose “what and where” anti-pooling technology [14] to restore the relative position information. Assuming that (c_x, c_y) is parameter of anti-pooling kernel, wherein the output size is $(H \times c_x, W \times c_y)$.
- ⑦ MLP: The fully connected layer which performs nonlinear fitting on high-order features. As shown in equation [4], we set input as x , weight as w , and offset as b , and the output of the full connection layer is as shown in the following formula:

$$h = f(w * x + b). \quad (4)$$

In this paper, f represents the activation function of each layer. Softplus function is used for mapping high-order features to hidden layer space z .

3 Methodology

3.1 Traffic preprocessing

The first few data messages in each session of encrypted traffic mainly include connection information (such as three-way handshake in TCP connection and key exchange in TLS connection) and a small amount of content exchange, which can better reflect the main features of the entire session. We preprocess the encrypted session into a special format image which is suitable for deep learning, and then utilize convolutional auto-encoding and convolutional neural network techniques to extract hidden layer features and classify images. The preprocessing of the data set, as shown in Fig. 3, mainly includes session segmentation, session filtering, extracting the packet byte matrix, transforming the image, and finally generating the ISVTD (Image set of SSL VPN Traffic Data) in IDX format [15].

3.1.1 Session preprocessing

We split each pcap format file from the original data set into multiple session files according to the same five tuple. Then, the duplicated files without application layer data and identical content are cleared. As shown in Fig. 4, every resulting pcap file contains 24 bytes of the file header, and the header of each packet in the pcap file contains 16 bytes. For TCP, the connection establishment requires three handshakes, and the connection release requires four handshakes, so at least seven packets are required. Each Ethernet frame has a minimum length of 64 bytes. Any frame less than 64 bytes in length is an invalid frame that is aborted due to collision.

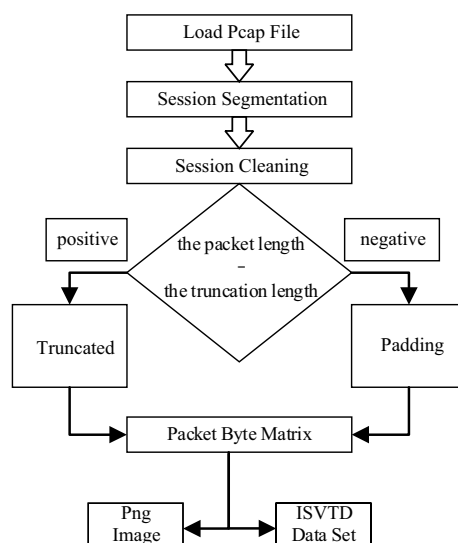


Fig. 3 Traffic preprocessing process

Each data set consists of three parts. S1, the first part, simulating all unmarked traffic samples, is used for unsupervised learning to discover hidden layer features. S2, the second part, a labeled training set, accounting for 90% of the total sample, is used for supervised classification training. Test, the third part, a labeled test set, accounting for 10% of the total sample, is used to evaluate the effect of the model.

3.2 CNN-based identification model

The CNN-based identification model [16] is shown in Fig. 7. It consists of two convolution group operations and two fully connected operations. The SGD is used to adjust the weight parameters, and finally, the relatively stable parameters are obtained. MLP continuously performs nonlinear fitting on the image to obtain better hidden layer features during the training process [17, 18]. The mapping relationship from the hidden layer feature to the specific classification is established from the tagged data set S2 during the experiment [19]. The mini-batch is resized to 50 and the training rounds are all 20,000 epochs. Stochastic Gradient Descent (SGD), being the optimization algorithm, is used to update weight:

$$\text{loss} = - \sum_{i=1}^n (\hat{y}_i \log y_i + (1 - \hat{y}_i) \log(1 - y_i)). \quad (5)$$

The loss function uses cross-entropy listed in formula [5]: y_i represents the expected output probability, \hat{y}_i represents

the actual output probability, and n represents the number of samples:

$$\hat{y}_i = \frac{\exp(z)}{\sum_1^n \exp(z)}. \quad (6)$$

As shown in formula [6], Softmax is the activation function. The number of samples is n , the input is z , and \hat{y}_i represents the confidence that the original input sample belongs to type i .

As shown in Table 4, the input image size is $39 \times 39 \times 1$. After the first operation of convolution group, the image size is changed to $20 \times 20 \times 32$, and the image size is changed to $10 \times 10 \times 64$ after the second operation of convolution group. The image finally becomes a 1024-dimensional hidden layer feature after the operation of flatten and full connection.

3.3 CAE-based identification model

3.3.1 Unsupervised pretraining phase

The unsupervised pretraining phase [20], as shown in Fig. 8, is based on convolutional auto-encoding. There are two neural networks, Encoder and Decoder. The encoder is responsible for extracting hidden layer features from the image samples. The decoder is responsible for reconstructing samples from the hidden layer features.

During the experiment, the hidden layer features are extracted from the unlabeled data set S_1 . The convolution operations and multi-layer perceptron weights were initialized with the Xavier method [21], making the approximately equal output variances of every layer. The mini-batch is sized to 100 and 200 epochs which are selected. The optimizer chooses the Adam algorithm, which iteratively updates the neural network weights using the training data. Minimizing the reconstruction error uses cross-entropy (binary cross-entropy) as the loss function, as shown in formula [5]. The specific parameters of the model architecture are shown in Table 5. The process includes four convolution group operations and deconvolution group operation, using

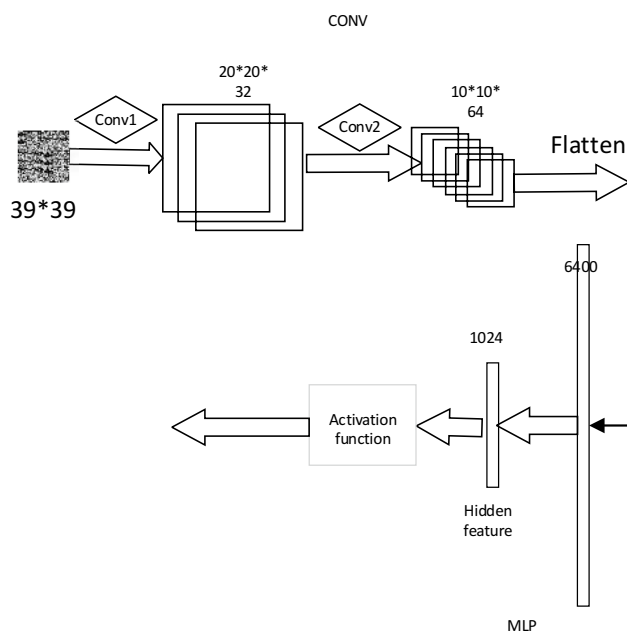


Fig. 7 CNN-based identification model

Table 4 Parameters of CNN-based model

Step	Operation	Parameter	Stride	Result size
Conv1	① Padding	Same		20×20×32
	③ Convolution	32×3×3	1	
	⑤ Pooling	2×2	2	
Conv2	① Padding	Same		10×10×64
	③ Convolution	64×3×3	1	
	⑤ Pooling	2×2	2	
MLP	⑦ Flatten	6400		1024
	⑦ MLP	1024		

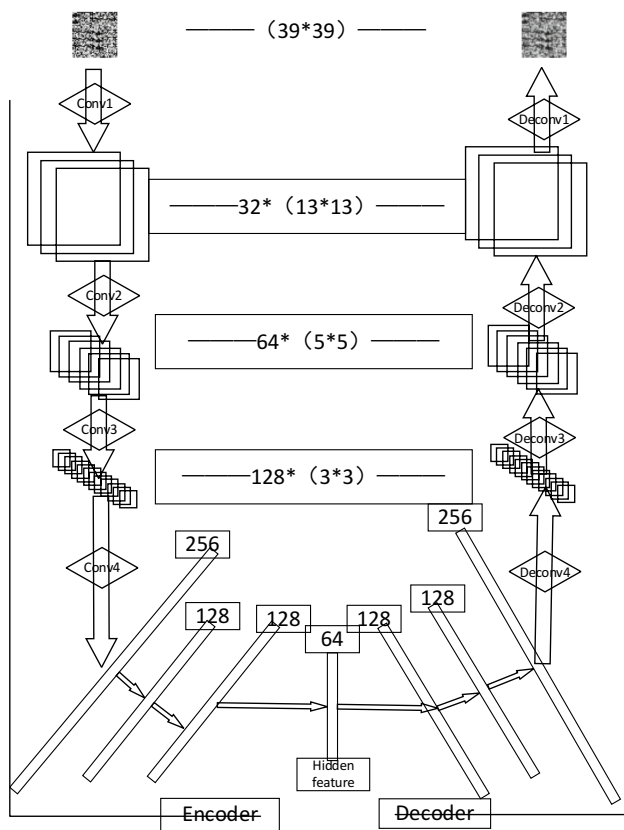


Fig. 8 Unsupervised pretraining process

six-layer MLP to map from high-order features to hidden layer features. Input image size is $39 \times 39 \times 1$ in the encoding network, transformed into $13 \times 13 \times 32$ after the first convolution group, $5 \times 5 \times 64$ after the second convolution group,

Table 5 Parameters of CAE-based model

Step	Operating	Parameter	Step	Operating	Parameter
Conv1	①	2×2	Deconv1	②	2×2
	③	$32 \times 5 \times 5$		④	$32 \times 5 \times 5$
	⑤	3×3		⑥	3×3
Conv2	①	2×2	Deconv2	②	2×2
	③	$64 \times 3 \times 3$		④	$64 \times 3 \times 3$
	⑤	3×3		⑥	3×3
Conv3	①	3×3	Deconv3	②	3×3
	③	$128 \times 3 \times 3$		④	$128 \times 3 \times 3$
	⑤	3×3		⑥	3×3
Conv4	①	1×1	Deconv4	②	1×1
	③	$256 \times 3 \times 3$		④	$256 \times 3 \times 3$
	⑤	3×3		⑥	3×3
MLP	⑦	256	MLP	⑦	256
	⑦	128		⑦	128
	⑦	128		⑦	128

$3 \times 3 \times 128$ after the third convolution group, and $1 \times 1 \times 256$ after the fourth convolution group. The image becomes a 64-dimensional hidden layer feature after three MLPs. The decoding network is the inverse of the encoding network.

3.3.2 Supervised classifier training phase

As shown in Fig. 9, the encoder network converts the high-order features of the conversation image into low-dimensional hidden layer features after the unsupervised pretraining process. We construct a classifier to learn with the tagged training data set, and map from the hidden layer features to output category, finally accomplishing the classification of the input.

The mapping relationship from the hidden layer feature to the specific classification is established from the tagged data set during the experiment. The optimization algorithm uses the Stochastic Gradient Descent (SGD) to update the weights and eventually obtain stable parameters. In the two-category identification, the number of training rounds is 2000 epochs, the loss function is cross-entropy function (binary cross-entropy) from the formula [5], and the activation function is the multi-type regression Sigmoid from equation [7]. Where z is the input and \hat{y} is the confidence of the classification result:

$$\hat{y} = \frac{1}{1 + e^{-z}}. \quad (7)$$

In the six-category identification, the number of training rounds is 2000 epochs, and the activation function is the normalized exponent function (Softmax) listed in formula [6]. The loss function is the multi-category cross-entropy function (categorical cross-entropy) listed in formula [8]:

$$\text{loss} = - \sum_{i=1}^n \hat{y}_{i1} \log y_{i1} + \hat{y}_{i2} \log y_{i2} + \dots + \hat{y}_{im} \log y_{im}. \quad (8)$$

The number of classification is m , the number of samples is n , y_{im} represents the expected output probability of the i th sample belonging to type m , and \hat{y}_{im} represents the actual output probability that the i th sample belongs to type m .

4 Experimental result

We choose Keras Library, which is based on TensorFlow, to implement our models. The specific experimental environment is shown in Table 6. Each of the proposed models was trained and evaluated against the independent test set that was extracted from the data set. This chapter first introduces the evaluation indicators, then verifies that the CAE-based identification model has the extraction ability of hidden layer feature, and finally tests the performance of the two models.

Fig. 9 Supervised classifier training process

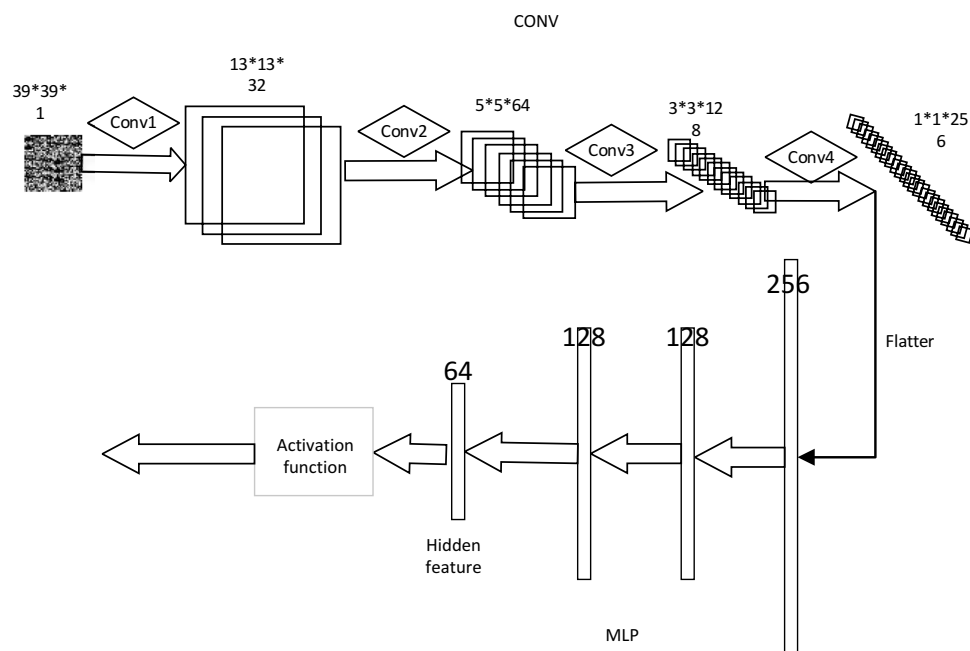


Table 6 Experimental environment parameter

Category	Parameter
Hardware platform	Dell XPS 8930 desktop computer
processor	6-core Intel Core i7-8700K
RAM	16 g memory
Graphics card	Nvidia GeForce GTX 1080
Deep learning platform	Keras
Deep learning backend	TensorFlow1.12
Cuda version	8.0
CuDNN version	8.0

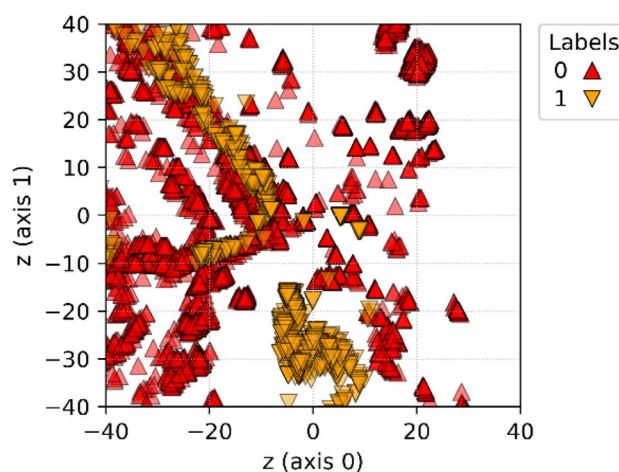


Fig. 10 Two-category hidden layer feature distribution

4.1 Evaluation indicators

Acc (Accuracy), Pre (Precision), Rec (Recall), and F1 (F1-Score) are four widely used metrics in statistical classification. Acc represents the overall effect of the method, and Pre, Rec, and F1 determine the identification effect of a certain type of traffic (Table 7).

Pre indicates how much the identified traffic is accurate and measures the accuracy of the identification system. Rec indicates how many of the correct entries have been identified, measuring the recall rate of the identification system. Sometimes, there are contradictions between the Pre and Rec indicators. To evaluate the advantages and disadvantages of different models, the concept of F1 values, as shown in formula [9], is proposed on the basis of Pre and Rec to make an overall evaluation of Precision and Recall:

$$F_1 = \frac{2 \times \text{Pre} \times \text{Rec}}{\text{Pre} + \text{Rec}} \times 100\%. \quad (9)$$

4.2 Hidden layer feature visualization

Due to the high-dimensional feature of the hidden layer feature of CNN-based models [22], the result is inconvenient for visualizing; we only adjust the hidden layer features of the CAE-based model, and then visualize it under two-dimensional conditions to observe the neighbor distribution of different categories [23]. Two-category traffic visualization is shown in Fig. 10. Six-category traffic visualization is shown in Fig. 11.

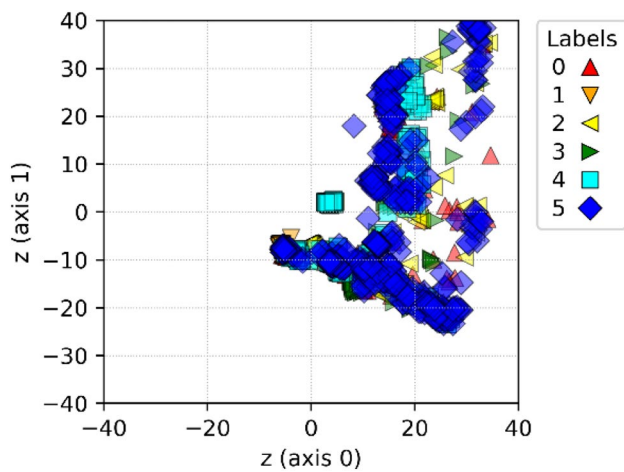


Fig. 11 Six-category hidden layer feature distribution

Table 7 Evaluation index confusion matrix

Confusion matrix	Predictive value	
	Positive	Negative
Actual value		
Positive	TP	FN
Negative	FP	TN
Accuracy (Acc)	Precision rate (Pre)	Recall rate (Rec)
$(TP + TN) / (TP + FP + TN + FN)$	$TP / (TP + FP)$	$TP / (TP + FN)$

In the above figure, each classification flow uses different color and shape to facilitate differentiation. Clusters of similar positions represent the similar structure and characteristics. It can be seen that the VPN and Non-VPN traffic have satisfactory discrimination, but the results of six different application categories included in the VPN traffic overlap more seriously and have minor separation distance. In particular, Voip (labeled 5) traffic is stacked with other traffic. The reason is the sample size of the ISVTD1 data set is big enough for the learning of unsupervised pretraining generation model, while the ISVTD2 data set has fewer samples and is unevenly distributed. Voip traffic, with insufficient unsupervised pretraining, accounts for more than 1/3 of all traffic. The hidden layer feature extraction of Voip traffic is not obvious enough.

4.3 Traffic identification result

The CAE-based model results, as shown in Table 8, are consistent with the visual invisible features, the two-category identification results are better than the six-category identification results, the overall accuracy of the two-category is 99.87%, and the overall accuracy of the six-category is

Table 8 CAE-based classification and identification result

Category (%)	Pre (%)	Rec (%)	F1 (%)
ISVTD1			
0 (non-VPN)	99.84	100	99.92
1 (VPN)	100	99.59	99.79
Acc of overall	99.87		
ISVTD2			
0 (chat)	74.07	86.96	79.99
1 (email)	92.86	100	96.29
2 (file transfer)	78.57	89.19	83.54
3 (P2P)	92.31	100	96
4 (streaming)	88.10	88.10	88.1
5 (Voip)	100	100	100
Acc of overall	90		

Table 9 CNN-based classification and identification result

Category (%)	Pre (%)	Rec (%)	F1 (%)
ISVTD1			
0 (non-VPN)	99.77	99.92	99.84
1 (VPN)	99.58	98.75	99.16
Acc of overall	99.74		
ISVTD2			
0 (chat)	78.57	81.48	79.99
1 (email)	100	100	100
2 (file transfer)	89.47	80.95	84.99
3 (P2P)	100	92.30	96
4 (streaming)	93.02	95.23	94.11
5 (Voip)	95.69	100	97.79
Acc of overall	92.92		

90%. In the specific application category identification of VPN [24], Voip traffic identification rate is the highest; F1-Score is 100%, followed by Email traffic which F1-Score is 96.29%. Chat traffic identification rate is the lowest; F1-Score is 79.99%.

The CNN-based model results, as shown in Table 9, have better two-category classification results than the six-category. The overall accuracy of the two-category is 99.74%, and the overall accuracy of the six-category is 92.92%. In the specific application category identification of VPN, Email traffic identification rate is the highest, F1-Score is 100%, followed by Voip traffic, F1-Score is 97.79%, Chat traffic identification rate is the lowest, and F1-Score is 79.99%.

By analyzing the classification results, we know that it is feasible to use the convolution method to extract the hidden features from the session image. The identification rate of the Non-VPN session is higher than the VPN session. The session identification rate for different application types in the six categories is as follows:

Table 10 Comparison of two-category identification result

Model	Non-VPN			VPN		
	Pre (%)	Rec (%)	F1 (%)	Pre (%)	Rec (%)	F1 (%)
KNN [4]	84.6	85.2	84.89	84.8	86.7	85.74
C4.5 [4]	90.6	88.8	89.69	89	92	90.48
1D-CNN [9]	100	99.9	99.94	99.9	100	99.94
CNN	99.77	99.92	99.84	99.58	98.75	99.16
CAE	99.84	100	99.92	100	99.59	99.79

Table 11 Comparison of the six-category identification result

Model	VPN		
	Pre (%)	Rec (%)	F1-score (%)
C4.5 [4]	84	87.6	85.76
1D-CNN [9]	94.9	97.3	96.08
CNN	92.79	91.66	92.22
CAE	87.65	94.04	90.73

Voip&Email > P2P > Streaming > File Transfer > Chat.

The reason may be that the Voip and Email formats are fixed, and the hidden layer features are highly differentiated. Chat traffic comes from a variety of different applications such as ICQ, AIM, Skype, Facebook, and Hangouts. The session architecture generated by different applications is quite different, and the hidden layer features are lowly differentiated [25].

5 Discussion

We compare the two-category identification results of the proposed model on the ISCX VPN2016 data set with the KNN and C4.5 models in [4], and the best result from 1D-CNN models in [9]. We only compare the six-category results with the C4.5 and 1D-cnn model as the six-category results of KNN model are not listed in the original literature [4].

As shown in Table 10, in the two-category traffic identification model, the CAE-based identification model is slightly superior to the CNN-based identification model. The results of both models are better than KNN and C4.5 methods. It is not much different from the current best results of literature [9]. As shown in Table 11, the CNN-based identification model is superior to the CAE-based identification model in the six-category traffic identification. The F1-Score of the CNN-based identification model in six-category is 1.49% higher than CAE-based identification model. The results of both models are better than the best results in the literature

[4], although there are still some gaps compared with the literature [9]. Due to the unsupervised features, the CAE-based model uses fewer tags than 1D-CNN model. The above results show that the CAE-based unsupervised identification model is suitable for large sample size, and the CNN-based supervised identification model is suitable for small sample size in the VPN traffic identification.

6 Conclusion

This paper proposes two deep learning-based methods using convolutional auto-encoding and convolutional neural network to classify and identify VPN encrypted traffic in real time [26]. As far as we know, this is the first time that a CAE-based identification model has been used in VPN traffic identification. This model utilizes the unsupervised features of CAE and the advantages of multi-layer perceptron in data dimensionality reduction, combining supervised classification learning technique, to achieve accurate identification of VPN encrypted traffic. The experimental results show that the best result in the classification of VPN and Non-VPN traffic comes from the CAE-based identification model, and the overall identification accuracy rate can reach 99.87%. Among the classification of different application types in VPN traffic, the best result comes from the CNN-based identification model, and the overall identification accuracy rate can reach 92.92%. In the future work, we will further study the unsupervised methods to identify other encrypted traffic and take the impact of data set size and balance on classification into consideration.

References

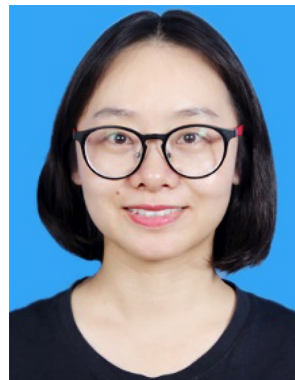
1. Wubin, P., Guang, C., Xiaojun, G., et al.: Review and perspective on encrypted traffic identification research[J]. Trans. Commun. **37**(9), 154–167 (2016) (in Chinese)
2. Qi, L., Zhou, Z., Jiguo, Yu., Liu, Q.: Data-sparsity tolerant web service recommendation approach based on improved collaborative filtering. IEICE Trans. Inf. Syst. **E100D**(9), 2092–2099 (2017)
3. Wei, W., Zhang, H., Li, B., et al.: Active Identification of VPN server based on correlation detecting[J]. Ind. Control Comput. **30**(3), 111–112 (2017) (in Chinese)

4. Draper-Gil, G., Lashkari, A.H., Mamun, M.S.I., et al.: Characterization of encrypted and VPN traffic using time-related features. In: Proceedings of the 2nd International Conference on Information Systems Security and Privacy (ICISSP 2016), pp. 407–414 (2016)
5. Bagui, S., Fang, X., Kalaimannan, E., et al.: Comparison of machine-learning algorithms for classification of VPN network traffic flow using time-related features. *J. Cyber Secur. Technol.* **1**(2), 108–126 (2017)
6. Yamansavascular, B., Guvensan, M.A., Yavuz, A.G., et al.: Application identification via network traffic classification. In: IEEE International Conference on Computing, Networking and Communications (ICNC). IEEE, pp. 843–848 (2017)
7. Wang, Z.: The applications of deep learning on traffic identification[J]. *BlackHat USA*, 24p (2015)
8. Lotfollahi, M., Siavoshani, M.J., Zade, R.S.H., et al.: Deep packet: a novel approach for encrypted traffic classification using deep learning. *Soft. Comput.* (2017). <https://doi.org/10.1007/s00500-019-04030-2>
9. Wang, W., Zhu, M., Wang, J., et al.: End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In: 2017 IEEE International Conference on Intelligence and Security Informatics (ISI). IEEE, pp. 43–48 (2017)
10. Li, D., Zhu, Y., Lin, W.: Mobile app traffic identification based on visual perception feature[J]. *J. Comput. App.* **2019**(4) (in Chinese)
11. Chen, X., Wang, P., Yu, J.: CNN based encrypted traffic identification method. *J. Nanjing Univ. Posts Telecommun. Nat. Sci. Edn.* (2018). <https://doi.org/10.14132/j.cnki.1673-5439.2018.06.006>
12. Wang, P., Chen, X.: SAE-based encrypted traffic identification method. *Comput. Eng.* **44**(11), 140–147 (2018). <https://doi.org/10.19678/j.issn.1000-3428.0052059>
13. Wang, W., Zhu, M., Zeng, X., et al.: Malware traffic classification using convolutional neural network for representation learning. In: 2017 International Conference on Information Networking (ICOIN). IEEE, pp. 712–717 (2017)
14. Jia, Q., Wang, X., Zhou, L., et al.: New Local feature description algorithm based on improved convolutional auto-encoder[J]. *Comput. Eng. Appl.* **53**(19), 184–191 (2017) (in Chinese)
15. Zhao, J., Mathieu, M., Goroshin, R., et al.: Stacked what-where auto-encoders[J] (2015). <http://arXiv.org/abs/1506.02351>
16. Xu, F., Zhang, X., Xin, Z., et al.: Investigation on the Chinese text sentiment analysis based on convolutional neural networks in deep learning[J]. *Comput. Mater. Contin.* **58**(3), 697–709 (2019)
17. Pan, L., Qin, J., Chen, H., et al.: Image augmentation-based food recognition with convolutional neural networks[J]. *CMC Comput. Mater. Contin.* **59**(1), 297–313 (2019)
18. Liu, Z., Xiang, B., Song, Y., et al.: An improved unsupervised image segmentation method based on multi-objective particle swarm optimization clustering algorithm[J]. *CMC Comput. Mater. Contin.* **58**(2), 451–461 (2019). (ISBN:978-1-4503-0000-0/18/06)
19. Hong, X., Zheng, X., Xia, J., et al.: Cross-lingual non-ferrous metals related news recognition method based on CNN with a limited bilingual dictionary[J]. *Comput. Mater. Contin.* **58**(2), 379–389 (2019)
20. Rezaei, S., Liu, X.: Deep learning for encrypted traffic classification: an overview. *IEEE Commun. Mag.* **57**(5), 76–81 (2019)
21. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics, pp 249–256 (2010)
22. Zhou, Z., Mu, Y., Wu, Q.M.J.: Coverless Image steganography using partial-duplicate image retrieval[J]. *Soft Comput.* **23**(13), 4927–4938 (2019)
23. Zhou, Z., Wu, J.Q.M., Sun, X.: Multiple distances-based coding: toward scalable feature matching for large-scale web image search. *IEEE Trans Big Data* (2019). <https://doi.org/10.1109/tbdata.2019.2919570>
24. Yildirim, T., Radcliffe, P.J.: VoIP traffic classification in IPsec tunnels. In: 2010 International Conference on Electronics and Information Engineering. IEEE, Vol 1, pp V1-151–V1-157 (2010)
25. Ximenes, E., Yeo, K.C., Azam, S., et al.: Performance analysis of various encryption techniques in communication network[J]. *Asian J. Inf. Technol.* **16**(1), 125–130 (2017)
26. Singh, K.K.V.V., Gupta, H.: A new approach for the security of VPN. In: Proceedings of the Second International conference on Information and Communication Technology for Competitive Strategies. ACM, 13p (2016)

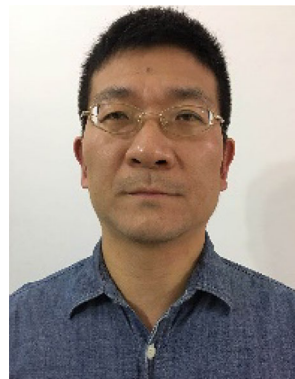
Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Lulu Guo received her B.S. and M.S. degrees from the State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou, China, in 2012 and 2019. His research concentrates on cyber security and machine learning. His email address is mdscience@sina.com.



Qianqiong Wu received the B.S. and M.S. degrees from Department of Automation, Faculty of Information Engineering, Nanchang University, China, in 2004 and 2007, respectively. She is currently a lecturer of Department of Electrical Engineering, Faculty of Electrical Engineering and Automation, Luoyang Institute of Technology, China. Her research interests are electrical engineering and deep learning.



Shengli Liu received the Ph.D. degree in computer science and engineering from State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou, China. Now, he is a professor at State Key Laboratory of Mathematical Engineering and Advanced Computing. His research interest includes network attack behavior detection, malicious code analysis, and network infrastructure security.



Ming Duan received the B.S. and M.S. degrees from the University of Information Engineering, Zhengzhou, China, in 2002 and 2005, respectively, and the Ph.D. degree in Shanghai Jiao Tong University, China in 2012. He is currently a professor of the State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou, China. His research interests are cryptanalysis and deep learning.



Jianwen Sun received the B.S. degree in Automation from Beijing Union University, China. She is currently pursuing the M.S. degree at the State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou, China. Her research interests are network traffic analysis technology and machine learning.



Huijie Li received his B.S. degree from the State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou, China, in 2010. He is currently pursuing the M.S. degree at the State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou, China. His research interests are cyber security and machine learning.