End-to-end Encrypted Traffic Classification with One-dimensional Convolution Neural Networks

Wei Wang, Ming Zhu
Department of Automation,
University of Science and Technology of China
Hefei, China
ww8137@mail.ustc.edu.cn, mzhu@ustc.edu.cn

Jinlin Wang, Xuewen Zeng, Zhongzhen Yang
National Network New Media Engineering Research Center,
Institute of Acoustics, Chinese Academy of Sciences
Beijing, China
{wangjl, zengxw, yangzz}@dsp.ac.cn

***Abstract*—Traffic classification plays an important and basic role in network management and cyberspace security. With the widespread use of encryption techniques in network applications, encrypted traffic has recently become a great challenge for the traditional traffic classification methods. In this paper we proposed an end-to-end encrypted traffic classification method with one-dimensional convolution neural networks. This method integrates feature extraction, feature selection and classifier into a unified end-to-end framework, intending to automatically learning nonlinear relationship between raw input and expected output. To the best of our knowledge, it is the first time to apply an end-to-end method to the encrypted traffic classification domain. The method is validated with the public ISCX VPN-nonVPN traffic dataset. Among all of the four experiments, with the best traffic representation and the fine-tuned model, 11 of 12 evaluation metrics of the experiment results outperform the state-of-the-art method, which indicates the effectiveness of the proposed method.**

Keywords—encrypted traffic classification; end-to-end; one-dimensional convolutional neural networks

I. INTRODUCTION

Traffic classification is the task of associating network traffic to a specific class according to the requirements, which has been a task of crucial importance in network management and cyberspace security. For example, in network management field, traffic can be classified based on the different priority so as to guarantee the quality of service (QoS) of the network. In cyberspace security field, traffic can be classified to benign traffic or malware traffic to achieve the goal of network anomaly detection. Recently, with the widespread use of encryption techniques in network applications, traffic encryption has become the standard practice nowadays. Especially, many malwares use encryption techniques such as TLS to encrypt the communication traffic to evade the detection of firewall and network intrusion detection system. Those practices bring new challenges to the traditional traffic classification methods [1].

According to the difference of ISO/OSI layer, traffic encryption techniques can be divided into application layer encryption, presentation layer encryption and network layer encryption [2]. Application layer encryption means that applications implement their own protocols for secure data transmission in application layer (e.g. BitTorrent or Skype), and it also called regular encryption in some papers. Presentation layer encryption and network layer encryption mean that applications encrypt the whole packets from the upper layer, and the typical techniques are TLS and IPsec, Some tunnel technology such as VPN is based on these techniques. This encryption type is also called protocol encapsulation. In some cases, encrypted traffic through regular encryption can be further encrypted through protocol encapsulation (e.g. Skype traffic through VPN). This paper focuses on both regular encrypted traffic classification and protocol encapsulated traffic classification.

According to the difference of granularity of requirements, encrypted traffic classification can be divided into encrypted traffic identification, encrypted traffic characterization and detailed encrypted traffic classification [3]. Encrypted traffic identification means identifying encrypted traffic from unencrypted traffic, and there are many studies about this problem. Detailed encrypted traffic classification means that associating traffic with a specific application. Due to the wide variety of applications and versions, this task is relatively difficulty. Encrypted traffic characterization means associating traffic with a type of application (e.g. chat or streaming), and many studies are carried out on this task recently [3] [4]. This paper focuses on encrypted traffic characterization.

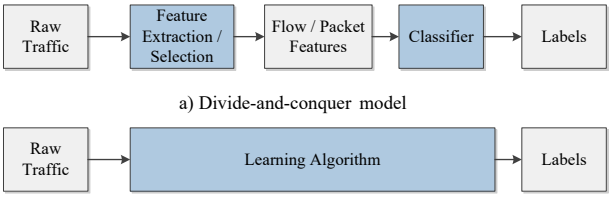


Fig. 1. Divide-and-conquer model vs end-to-end model

There are four main methods of traffic classification [5]: port-based, deep packets inspection (DPI)-based, statistical-based and behavioral-based. The accuracy of port-based method is very low nowadays because of the common use of random port and port disguise. DPI-based method encounters great difficulties because it cannot decrypt the encrypted traffic. The current research mainly focuses on statistical-based method and behavioral-based method. Based on the traffic classification taxonomy proposed by [6], they are both machine learning approaches, whose general workflow is as follows: firstly hand-designing the traffic features (e.g. flow features or packet features), secondly extracting and selecting those features form raw traffic, finally classifying traffic with those features by hand-designed classifier (e.g. decision tree or Naive Bayes). The workflow is processed step by step and each step is independent of other steps. Figure 1a shows the procedure described above. It is essentially a divide-and-conquer strategy [7], which decomposed a complex problem into several sub-problems. On one hand, the sub-problems become much simpler and more controllable. On the other

Supported by the "Strategic Priority Research Program" of the Chinese Academy of Sciences, Grant No. XDA06011203, XDA06010302.

hand, the disadvantage is also obvious. The optimal solution of sub-problem does not mean the optimal solution of the global problem, and the optimal solution of each step does not mean the optimal solution of the whole process.

In this paper, we propose an end-to-end method of encrypted traffic classification with one-dimensional convolution neural networks [8] (1D-CNN). This method is based on deep learning which is a representative technology in representation learning domain. 1D-CNN is applied as learning algorithm, and it directly learns features from raw traffic automatically. The traffic features are learned layer by layer, and the high level features are as the input of softmax layer. Finally the 1D-CNN directly outputs the predicted labels. Figure 1b shows the procedure described above. That strategy is the end-to-end strategy, which is commonly used in deep learning technology. Instead of dividing a complex problem into sub-problems, it can directly learn the nonlinear relationship between the raw traffic input and the expected output label. Compared with the divide-and-conquer strategy, end-to-end strategy has a natural synergistic effect [9] and it is more likely to get the global optimal solution.

The contributions of our work are as follows. Firstly, we propose an end-to-end encrypted traffic classification method with 1D-CNN, and to the best of our knowledge, this is the first time to apply an end-to-end method to the encrypted traffic classification domain. Secondly, we determine the best representation type of encrypted traffic and the best 1D-CNN model architecture by contrasting various experiments. Finally, our study yields significant improvements to the state-of-the-art method on a public ISCX VPN-nonVPN (ISCX) traffic dataset [10]. Besides, all the relevant research data, such as our training code, training data, fine-tuned model and detailed experiment results will be published on GitHub (https://github.com/echowei/cnn-traffic) for any researchers whoever is interested.

The rest of this paper is organized as follows. Section II describes related work. Section III describes the methodology of the proposed method using 1D-CNN. Section IV mainly covers the experiment results and analysis. Section V presents the discussion and future work. Section VI provides concluding remarks.

II. RELATED WORK

The current research of encrypted traffic classification mainly focuses on machine learning approach. There are two main types of features which are most frequently used by those researches [2]: flow features (e.g. duration per flow, flow bytes per second) and packet features (e.g. packet size, inter-packet duration of the first n packets). According to the review of Velen et al. [2] on 26 papers about encrypted traffic classification in the 2005-2015 period, 12 papers use flow features, while 5 papers use packet features, 7 papers use the combination of flow and packet features and 2 papers use other features.

There are many researches on regular encrypted traffic classification now. For example, Wang et al. [11], Coull et al. [12] and Mauro et al. [13] researched P2P, iMessage and WebRTC respectively. What they applied are flow features, packet features and flow features respectively. The corresponding classifiers are C4.5 decision tree, Naive Bayes and random forest respectively. Relatively speaking, fewer researches on protocol encapsulated traffic classification are conducted. For example, Aghaei et al. [14] proposed a classification method with flow features and C4.5 decision tree classifier on proxy traffic. Draper-Gil et al. [3] proposed a classification method with only time related flow features on both regular encrypted traffic and protocol encapsulated traffic. It is worth noting that they published a valuable dataset including both those two types of traffic.

The papers mentioned above all adopted traditional divide-and-conquer strategy, and the number of papers using end-to-end strategy to perform traffic analysis is very small now. Wang [15] proposed a stacked auto encoder (SAE) based network protocol identification method. Our team proposed a malware traffic classification method with 2D-CNN [6]. Those two papers both used raw traffic as input and final labels as output. From this point of view, they are end-to-end methods. Gao et al. [16] and Javaid et al. [17] applied deep belief networks (DBN) and sparse auto encoder (SAE) to research network traffic classification respectively. While those two techniques are deep learning technology, they used hand-designed features other than raw traffic as input. That is to say, their methods are not end-to-end methods.

III. METHODOLOGY

A. DataSet

A lot of researches about encrypted traffic classification used self-collected traffic or private traffic of security companies, damaging the credibility of their results. According to the review of Velen et al. [2] on 26 papers about encrypted traffic classification in the 2005-2015 period, only 6 papers used public dataset. Because classic machine learning approaches need hand-designed features as input, many current public traffic datasets are features datasets other than raw traffic datasets, e.g. datasets used by 6 papers in the review mentioned above. Draper-Gil et al. [3] published ISCX dataset which includes 7 types of regular encrypted traffic and 7 types of protocol encapsulated traffic. These applications are the most popular nowadays and the types are also very diverse. There are two data formats in this traffic dataset, flow features and raw traffic (i.e. pcap format). We used ISCX dataset as our test dataset.

The flow features of ISCX dataset have 14 classes of labels, but the raw traffic has no labels, so we labeled pcap files in the dataset according to the description of their paper. Some files such as “Facebook_video.pcap” can be labeled as either “Browser” or “Streaming”, and all files related to “Browser” and “VPN-Browser” all have this problem. We can’t solve this problem even after email communication with the authors, so we decided not to label these files. Finally the labeled ISCX dataset has 12 classes, including 6 classes of regular encrypted traffic and 6 classes of protocol encapsulated traffic. Table I shows the detailed content of labeled ISCX dataset.

TABLE I. LABELED ISCX VPN-NONVPN DATASET

Traffic Type	Content
Email	Email、Gmail (SMPT, POP3,IMAP)
VPN-Email	
Chat	ICQ, AIM, Skype, Facebook, Hangouts
VPN-Chat	
Streaming	Vimeo, Youtube, Netflix, Spotify
VPN-Streaming	
File transfer	Skype, FTPS, SFTP
VPN-File transfer	
VoIP	Facebook, Skype, Hangouts, Voipbuster
VPN-VoIP	
P2P	uTorrent, Bittorrent
VPN-P2P	

B. End-to-end Framework Overview

Figure 2 shows the overview of our proposed end-to-end encrypted traffic classification method, consisting of preprocess phase, training phase and test phase. Compared with the traditional machine learning methods using divide-and-conquer, this framework does not contain these independent modules such as feature extraction, feature selection and classifier. In fact these modules are integrated into the CNN model. The features are automatically learned and the traffic is directly classified through the softmax layer, and then the nonlinear relationship between the raw input and the expected output are determined, achieving the goal of end-to-end learning.

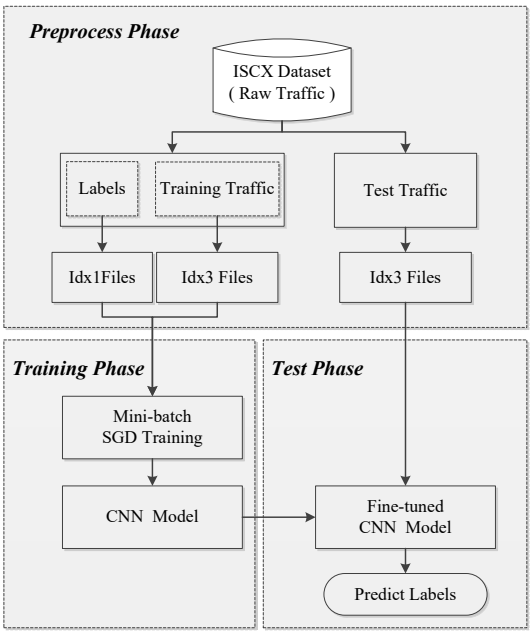


Fig. 2. End-to-end Framework Overview

Phase I (Preprocess Phase). This phase preprocesses the raw traffic of ISCX dataset and generates the required format for the input data of CNN model. The preprocess tool is USTC-TL2016 developed by our team in [6], consisting of four steps: traffic split, traffic clean, image generation and IDX conversion. Please refer to our paper for detailed description about this tool. It should be noted that the content of the generated IDX3 files is byte data of raw traffic, rather than flow features or packet features, which is different from traditional divide-and-conquer method.

Phase II (Training Phase). This step trains the CNN model with IDX3 traffic data and IDX1 label data generated from phase I. The training method is mini-batch stochastic gradient descent (SGD). 10-fold cross-validation technique is used to guarantee the generalization ability of CNN model. The result hyper parameters will be used in test phase.

Phase III (Test Phase). This step predicts the class label of IDX3 traffic data generated from phase I with the fine-tuned CNN model trained in phase II, and finally get the classification results.

There are two problems that require further study and discussion. The first problem is the traffic representation choices. In other words, which part of raw traffic will be used and how to organize these traffic byte data? The second problem is CNN model choices. In other words, what kind of CNN model is more effective? Those two problems will be discussed in section C and D.

C. Traffic Representation

1) Representation choices

Firstly, the raw traffic needs to be split into discrete units. The most common two choices of traffic representation are session and flow [18]. A session is a traffic unit divided based on 5-tuple, i.e. source IP, source port, destination IP, destination port and transport-level protocol. A flow is very similar to a session, and the difference is that it contains traffic of only one direction, i.e. the source and destination IP / port are not interchangeable.

Secondly, the traffic byte data in each packet can be divided into multiple protocol layers. There are two types of layer choice in our work. The first choice is layer 7 in ISO/OSI model or layer 4 in TCP/IP model (L7). Intuitively the characteristics of traffic should be reflected in this layer. For example, STMP protocol represents email traffic and HTTP protocol represents browser traffic. They are both protocols of application layer Based on this assumption; Wang [15] only selects L7. The second choice is all protocol layers (ALL). Sometimes the traffic data under L7 also contain encrypted traffic feature information. For example, the initial security handshake of TLS is also under L7.

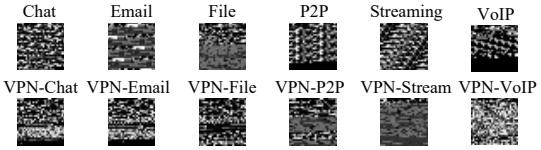
In summary, we studied four choices of traffic representation, including Session + L7, Session + All, Flow + L7 and Flow + All. It should be noted that different flows or sessions may have different size, but the input data size of CNN must be uniform, so only first n bytes (n = 784 in this paper) of each flow or session are used.

2) Preprocessed Result

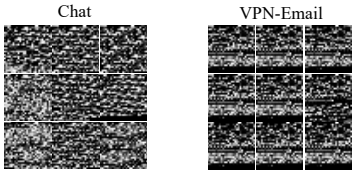
We preprocess ISCX dataset and get four groups of results. Table II shows these results. The preprocessed results can be analyzed with visualization technology. The visualization results of Session + All are shown in Figure 3, and the size is 784 bytes. It is obvious that different classes of traffic have obvious discrimination degree and each class of traffic has high consistency. It is reasonable to presume that our approach can achieve good performance.

TABLE II THE PREPROCESSED RESULTS OF ISCX DATASET

Representation	Traffic	Samples	Total
Session + All	Non-VPN	26921	38816
	VPN	12525	
Session + L7	Non-VPN	20173	32437
	VPN	12264	
Flow + All	Non-VPN	32422	50348
	VPN	17926	
Flow + L7	Non-VPN	22716	38723
	VPN	16007	
Total	---	---	160324



a) Visualization of all classes of traffic



b) Consistency in the same traffic class

Fig. 3. Visualization of Encrypted Traffic

D. CNN Model

1) Why Applying 1D CNN

So far, CNN has been mainly applied in the domain of computer vision, for example, image classification. Recently, there are some successful applications in the domain of natural language process (NLP) [19]. In fact, According to LeCun's research [20], CNN is suitable for the following kinds of data: data in the form of multiple array; data that have strong local correlations; data whose features can appear anywhere; data in which objects are invariant to translations and distortions. Specifically, 1D-CNN is good for data like sequential data or language. 2D-CNN is good for data like images or audio spectrograms. 3D-CNN is good for data like video or volumetric images.

Recently, there are some researches about applying CNN in network traffic analysis, such as malware classification by our team [6]. It transformed traffic to two dimensional images like Figure 3, and then applied 2D-CNN to classify the traffic images, achieving the goal of traffic classification. The method of Wang [15] is very similar and the only difference is that they used stacked auto encoder (SAE) other than 2D-CNN.

As far as we are considered, network traffic is essentially sequential data. It is a one dimensional byte flow organized by hierarchical structure. The structure of byte, packet, session and the whole traffic is very similar to the one of character, word, sentence and the whole article in the domain of NLP. In recent years, the successful applications of CNN in NLP all used 1D-CNN, for example, sentiment analysis and text classification [21][22]. In this paper, we are inspired by those studies to perform encrypted traffic classification task with 1D-CNN, and compare its performance with 2D-CNN.

2) Model Choices

The architecture of 2D-CNN model for comparison is the same with the architecture used in our previous work [6]. Please refer to that paper for detailed description. The architecture of 1D-CNN model is described below.

Let $x_i \in \mathbb{R}$ be the k-dimensional vector correspoding to the i-th traffic byte in the session or flow. A session or flow of length n is represented as :

$$x_{1:n} = x_1 \oplus x_2 \oplus ... \oplus x_n,$$

where $\oplus$ is the concatenation operator. In general, let $x_{i:i+j}$ refer to the concatenation of traffic bytes $x_i, x_{i+1}, ..., x_{i+j}$. A convolution operation involves a filter $w \in \mathbb{R}$, which is applied to a window of h traffic bytes to produce a new feature. For example, a feature c_i is generated by

$$c_i = f(w \cdot x_{i:i+h-1} + b).$$

Here $b \in \mathbb{R}$ is a bias term and f is ReLUs [23]. This filter is applied to each possible window of traffic bytes $\{x_{1:h}, x_{2:h+1}, ..., x_{n-h+1:n}\}$ to produce a feature map

$$c = [c_1, c_2, ..., c_{n-h+1}],$$

with $c \in \mathbb{R}$. We then apply a max-over-time pooling operation [24] over the feature map and take the maximum value $\hat{c} = \max\{c\}$ as the next feature.

We use multiple convolution layers and multiple pooling layers to extract high level features. These features form the penultimate layer and are passed to a fully connected softmax layer whose output is the probability distribution of the input session or flow.

It should be noted that both the convolution filter and max-over-time pooling are one dimensional operation which is the key characteristic of 1D-CNN. Table III describes the main parameters of each layer in our 1D-CNN model.

TABLE III THE MAIN PARAMETERS OF 1D-CNN MODEL

Layer	Operation	Input	Filter	Stride	Pad	Output
1	conv+ReLU	784*1	25*1	1	same	784*32
2	1d max pool	784*32	3*1	3	same	262*32
3	conv+ReLU	262*32	25*1	1	same	262*64
4	1d max pool	262*64	3*1	3	same	88*64
5	full connect	88*64	--	--	none	1024
6	full connect	1024	--	--	none	2/6/12
7	softmax	2/6/12	--	--	none	2/6/12

E. Experiments of Comparison

We compare the experiment results on ISCX dataset with the results of the state-of-art method in [3] to validate the performance of our proposed method. There are four experiments and they all have contrastive experiment in [3]. Table IV describes those experiments.

TensorFlow [25] is used as software framework which runs on Ubuntu 14.04 64bit OS. Server is DELL R720 with 16GB memory. An Nvidia Tesla K40m GPU is used as accelerator. 1/10 of data were randomly selected as test data, the rest is training data. The mini-batch size is 50 and the cost function is cross entropy. Gradient descent optimizer built in TensorFlow is used as optimizer. The learning rate is 1.0e-4, training time is about 40 epochs.

TABLE IV CONTRASTIVE EXPERIMENTS DESCRIPTION

Exp	Description	Classifier	Compared Exp
1	Protocol encapsulated traffic identification	2-class	A1
2	Regular encrypted traffic classification	6-class	A2
3	Protocol encapsulated traffic classification	6-class	A3
4	Encrypted traffic classification	12-class	B

IV. EVALUATION

A. Evaluation Metrics

Three evaluation metrics are used: accuracy, precision, recall. Accuracy is used to evaluate the overall performance of a classifier. Precision and recall are used to evaluate the performance of every class of traffic. TP is the number of instances correctly classified as X, TN is the number of instances correctly classified as Not-X, FP is the number of instances incorrectly classified as X, and FN is the number of instances incorrectly classified as Not-X.

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

$$precision = \frac{TP}{TP + FP}, recall = \frac{TP}{TP + FN}$$

B. Traffic Representation Evaluation

TABLE V THE ACCURACY OF DIFFERENT REPRESENTATIONS (%)

Experiment Accuracy				Which is better			
		All	L7	Session	Flow	All	L7
Exp 1	Session	99.9	95.4	2	0	2	0
	Flow	99.9	92.3				
Exp 2	Session	81.7	83.0	1	1	0	2
	Flow	81.8	82.5				
Exp 3	Session	98.3	95.3	1	1	2	0
	Flow	98.6	93.0				
Exp 4	Session	86.6	83.7	2	0	2	0
	Flow	86.5	81.0				
Total	---			**6**	**2**	**6**	**2**

Table V shows the experiment results of different types of traffic representation with 1D-CNN.

Session vs Flow. We can see from the table above, there are 6 comparisons which show that session is better than flow, and the accuracy is 1.45% higher on average. Only 2 comparisons show that flow is better than session, and the accuracy is only 0.2% higher. We can conclude that the session is more suitable as the type of traffic representation for encrypted traffic classification. An intuitive explanation can be given. It is because session contains bidirectional flows and contains more interaction information than unidirectional flow that the end-to-end method can learn more features from session than flow.

All layers vs L7. The table above also shows that there are 6 comparisons which show that all layers are better than L7, and the accuracy is 4.85% higher on average. Only 2 comparisons show that L7 is better than all layers, and the accuracy is only 1.0% higher on average. We can conclude that the type of all layers is more suitable as traffic representation for encrypted traffic classification. An intuitive explanation can be given for that conclusion. The type of all layers contains all data under application layer, and reserves more information of the phase of encryption negotiation. For example, the negotiation of cipher suite of TLS is performed on transport layer. The end-to-end method can learn more representative features from all layers.

In summary, the best type of traffic representation is session + all layers, and this conclusion is consistent with our previous findings of [6]. Only this type of traffic representation will be adopted when we perform CNN model evaluation.

C. CNN Model Evaluation

Figure 4 shows the comparison about the accuracy of 1D-CNN and 2D-CNN. Figure 5 and 6 show the precision comparison and the recall comparison of 12 classes of encrypted traffic in Exp 4. The first three experiments show similar pattern.

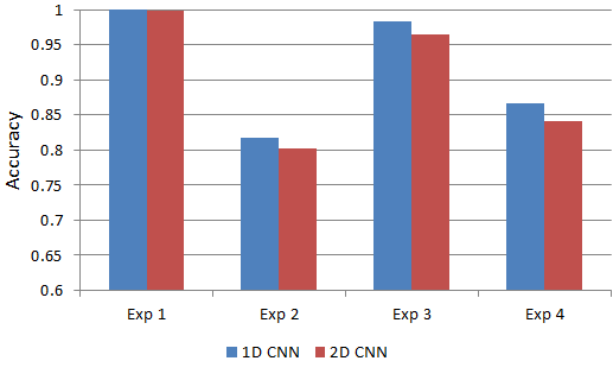


Fig. 4. The Accuracy Comparison of 1D-CNN and 2D-CNN

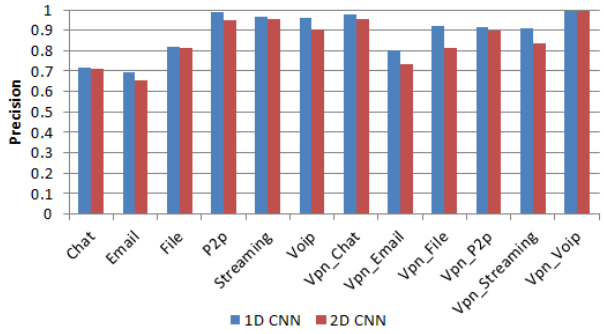


Fig. 5. The Precision comparison of 1D-CNN and 2D-CNN (Exp 4)

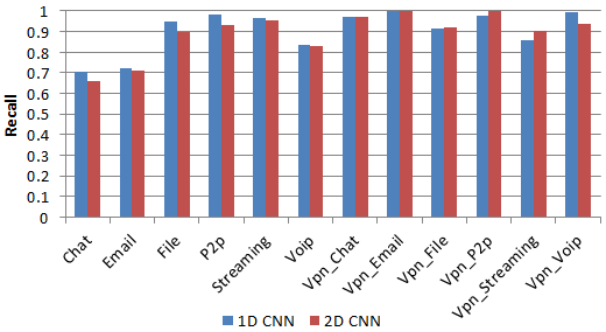


Fig. 6. The Recall Comparison of 1D-CNN and 2D-CNN (Exp 4)

As seen in Figure 4, the accuracy of two CNN models is higher than 80%. The accuracy of 1D-CNN is higher than 2D-CNN, as much as 2.51%. As seen in Figure 5, among all 12 classes of encrypted traffic in Exp 4, the precision of 11 classes of 1D-CNN is higher than 2D-CNN, 3.75% higher on average. As seen in Figure 6, among all 12 classes of encrypted traffic in Exp 4, the recall of 8 classes of 1D-CNN is higher than 2D-CNN, 2.96% higher on average. Besides, there are two classes that their recall is approximately equal.

In summary, the 1D-CNN has better performance than 2D-CNN on the task of encrypted traffic classification. It proves our assumption that 1D-CNN is more suitable to one dimensional traffic data. By contrast, the 2D-CNN's advantage of learning two dimensional spatial features is not obvious when one dimensional encrypted traffic classification is performed, so its performance is relatively lower than 1D-CNN.

D. Performance Comparison with the State-of-the art

Table VI, VII and VIII show the performance comparison on ISCX dataset between our proposed method and the state-of-the-art method in [3]. Their paper only showed the precision and recall results, so we compare performance only using those two evaluation metrics. The precision and recall of Table VII and VIII are average values. Please refer to our shared Github files for detailed experiment results.

TABLE VI PERFORMANCE COMPARISON (EXP 1, %)

	Non-VPN		VPN	
	Precision	Recall	Precision	Recall
C4.5	90.6	88.8	89	92
1D CNN	100	99.9	99.9	100
Improvement	9.4	10.1	10.9	8

TABLE VII PERFORMANCE COMPARISON (EXP 2 & 3, %)

	Non-VPN		VPN	
	Precision	Recall	Precision	Recall
C4.5	89	85.5	84	87.6
1D CNN	85.5	85.8	94.9	97.3
Improvement	-3.5	0.3	10.9	9.7

TABLE VIII PERFORMANCE COMPARISON (EXP 4, %)

	Non-VPN		VPN	
	Precision	Recall	Precision	Recall
C4.5	84.3	79.3	78.2	81.3
1D CNN	85.8	85.9	92	95.2
Improvement	1.5	6.6	13.8	13.9

1D-CNN achieves outstanding performance in Exp 1. The precision is 9.4% and 10.9% higher on Non-VPN and VPN traffic respectively than the state-of-the-art method. The recall is 10.1% and 8% higher on Non-VPN and VPN traffic respectively than the state-of-the-art method. 1D-CNN also achieves outstanding performance on VPN traffic in Exp 2, 3 and 4. The precision is 10.9% and 13.8% higher than the

state-of-the-art method, and the recall is 9.7% and 13.9% higher than the state-of-the-art method. 1D-CNN achieves good performance on Non-VPN traffic in Exp 4. The precision and recall is 1.5% and 6.6% higher respectively than the state-of-the-art method.

The performance of 1D-CNN on Non-VPN traffic in Exp 2 and 3 is not very good. Although the precision gets an improvement of 0.3%, the recall is 3.5% lower than the state-of-the-art method. That shows that there is a great performance difference between Non-VPN traffic and VPN traffic. After analyzing the images of those two types of traffic, we find that the discrimination degree of VPN traffic seem to be better than Non-VPN traffic. The detailed reason is to be investigated in the future work.

In summary, all four experiments achieve greater improvement than the state-of-the-art method. It confirms the effectiveness of our proposed end-to-end encrypted traffic classification method with 1D-CNN.

V. DISCUSSION

Our proposed end-to-end encrypted traffic classification method can leave out traditional steps, such as feature design, features extraction and features selection which are commonly used in traditional divide-and-conquer method. It uses 1D-CNN to automatically learn more representative features of encrypted traffic. The experiment results obtain better performance than the state-of-the-art method. There are three problems to be further studied in the future work. Firstly, about traffic representation, we use the first 748 bytes of each session as raw traffic. Because different classes of traffic have different types of packets, the more appropriate byte number needs to be further studied. Secondly, about model training, the current training data is imbalanced which has a great effect on experiment performance. For example, VPN-VoIP traffic has 6000 training samples and VPN-Email has only 298 training samples, and their precision is 99.5% and 80% respectively. We plan to study how to improve the 1D-CNN performance when training data is imbalanced. Thirdly, the experiment results show that Non-VPN traffic has relatively worse performance. We plan to analyze the detail reason and make corresponding improvement.

VI. CONCLUSION

On the basis of analysis of traditional encrypted traffic classification method using divide-and-conquer strategy in the domain of machine learning, a new end-to-end encrypted traffic classification method with 1D-CNN was proposed in this paper. The method integrates feature design, feature extraction and feature selection into a single framework, and can automatically learn more representative traffic features. Compared to the divide-and-conquer strategy, end-to-end strategy has a natural synergistic effect and it is more likely to get the global optimal solution. The best type of traffic representation was determined through experiments. We found that 1D-CNN is more suitable to the task of encrypted traffic classification than 2D-CNN, for network traffic is essentially one dimensional sequential data which can make full use of the advantage of 1D-CNN. The experiment results on public ISCX dataset yielded significant improvements to the state-of-the-art method, validating the effectiveness of our proposed end-to-end method. Both this work and our previous work [6] have proved that deep learning technology such as CNN shows good potential on the domain of traffic classification. We will continue to conduct in-depth research in this direction.

REFERENCES

[1] Z. Cao, G. Xiong, Y. Zhao, Z. Li and L. Guo, "A survey on encrypted traffic classification" in Applications and Techniques in Information Security, Springer, pp. 73-81, 2014.

[2] P. Velan, M. Cermak, P. Celeda and M. Drasar, "A survey of methods for encrypted traffic classification and analysis", International Journal of Network Management, vol. 25, no. 5, pp. 355-374, 2015.

[3] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun and A. A. Ghorbani, "Characterization of encrypted and VPN traffic using time-related features", In Proceedings of the 2nd International Conference on Information Systems Security and Privacy(ICISSP), pp. 407-414, 2016.

[4] X. Che, B. Ip and L. Lin, "A Survey of Current YouTube Video Characteristics," in IEEE MultiMedia, vol. 22, no. 2, pp. 56-63, Apr.-June 2015.

[5] E. Biersack, C. Callegari and M. Matijasevic, Data traffic monitoring and analysis. Berlin: Springer, 2013.

[6] W. Wang, X. Zeng, X. Ye, Y. Sheng and M. Zhu, "Malware Traffic Classification Using Convolutional Neural Networks for Representation Learning" In the 31st International Conference on Information Networking (ICOIN), Accepted, 2017.

[7] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, Introductions to Algorithms, London, pp. 65-113, 2009.

[8] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, "Gradient-Based Learning Applied to Document Recognition", Proc. IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.

[9] G. E. Dahl T. N. Sainath and G. E. Hinton "Improving Deep Neural Networks for LVCSR Using Rectified Linear Units and Dropout " in Proc. ICASSP, 2013.

[10] ISCX VPN-nonVPN encrypted network traffic dataset. http://www.unb.ca/cic/research/datasets/vpn.html, 2017

[11] D. Wang, L. Zhang, Z. Yuan, Y. Xue and Y. Dong, "Characterizing application behaviors for classifying p2p traffic", Computing Networking and Communications (ICNC) 2014 International Conference on. IEEE, pp. 21-25, 2014.

[12] S. E. Coull and K. P. Dyer, "Traffic analysis of encrypted messaging services: Apple iMessage and beyond", ACM SIGCOMM Comput. Commun. Rev., pp. 5-11, 2014.

[13] M. D. Mauro and M. Longo, "Revealing encrypted WebRTC traffic via machine learning tools", SECRYPT 2015 - Proceedings of the 12th International Conference on Security and Cryptography, pp. 259-266, 20–22 July, 2015.

[14] V. Aghaei-Foroushani and A. Zincir-Heywood, "A proxy identifier based on patterns in traffic flows," in HASE, Jan 2015.

[15] Z. Wang, "The Applications of Deep Learning on Traffic Identification." https://goo.gl/WouIM6. 2015

[16] N Gao, L Gao, Q Gao, "An Intrusion Detection Model Based on Deep Belief Networks", Advanced Cloud and Big Data (CBD) 2014 Second International Conference on, pp. 247-252, 2014.

[17] A. Javaid, Q. Niyaz, W. Sun and M. Alam. "A Deep Learning Approach for Network Intrusion Detection System." in Proc.9th EAI International Conference on Bio-inspired Information and Communications Technologies. New York, 2016.

[18] A. Dainotti, A. Pescape and K. Claffy, "Issues and future directions in traffic classification", Network IEEE, vol. 26, no. 1, pp. 35-40, 2012.

[19] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy and B. Shuai, "Recent Advances in Convolutional Neural Networks", arXiv preprint arXiv:1512.07108, 2017.

[20] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning", Nature, vol. 521, pp. 436-444, May 2015.

[21] Y. Kim, "Convolutional Neural Networks for Sentence Classification", Proc. Empirical Methods Natural Language Processing, pp. 1746-1751, 2014.

[22] X. Zhang and Y. LeCun, "Text understanding from scratch", arXiv preprint arXiv:1502.01710, 2016.

[23] V. Nair and G.E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines", Proc. Int'l Conf. Machine Learning, 2010.

[24] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu and P. Kuksa, "Natural Language Processing (almost) from Scratch", J.Machine Learning Research, vol. 12, pp. 2493-2537, 2011.

[25] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, et al., "Tensor-Flow: Large-Scale Machine Learning on Heterogeneous Distributed Systems", arXiv preprint arXiv:1603.04467, 2016.