2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th Intl. Conference on Data Science and Systems

Encrypted Traffic Classification with a Convolutional Long Short-Term Memory Neural Network

Zhuang Zou[1,2], Jingguo Ge[1], Hongbo Zheng[1], Yulei Wu[3], Chunjing Han[1], Zhongjiang Yao[1,2]
[1]Institute of Information Engineering, Chinese Academy of Science, Beijing 100093, China
[2]School of Cyber Security, University of Chinese Academy of Science, Beijing 100049, China
[3]College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter, EX4 4QF, UK
Email:{zouzhuang, gejingguo, hanchunjing}@iie.ac.cn, y.l.wu@exeter.ac.uk

Abstract—**With the rapidly emerging encryption techniques for network traffic, the classification of encrypted traffic has increasingly become significantly important in network management and security. In this paper, we propose a novel deep neural network that combines both the convolutional network and the recurrent network to improve the accuracy of the classification results. The convolutional network is used to extract the packet features for a single packet. The recurrent network is trained to pick out the flow features based on the inputs of the packet features of any three consecutive packets in a flow. The proposed model surpasses the existing studies which ask for the first N packets of a flow, and it provides more flexibility in real practice. We compare our model with the existing work under deep learning for encrypted traffic classification, based on the public dataset. The experimental results show that our model outperforms the state-of-the-art work in terms of both higher efficiency and effectiveness.**

I. Introduction

Recently, the traffic encryption has been widely used in the Internet due to the advanced encryption techniques. The encryption techniques not only protect Internet user's freedom, privacy and anonymity, but also make them evade the detection by firewall and circumvent the surveillance systems. However, encryption techniques have also been used to get immoral profits by adversaries. For example, attackers encrypt the malware traffic to intrude and attack systems anonymously. In addition, criminals utilize the privacy-enhancing tool (e.g., Tor [1]) to penetrate the dark net, where they could buy and sell drugs, weapons, counterfeit documents (e.g., passports, driver's licenses, social security cards, and utility bills), and counterfeit money, as well as to provide a medium for contract killers to solicit clients [2]. That is, the abuse of encryption techniques brings a new threat to cyber security and network management [3]. The identification and classification of encrypted traffic are therefore attracted significant attentions from both academia and industry.

As a key technique for advanced network management, including traffic engineering and anomaly detection [4], network traffic classification has been dedicated tremendous efforts in researches and received remarkably satisfying results. Nevertheless, the rapid growth of encrypted network traffic poses additional challenges to traffic classification. The existing solutions of solving the traffic classification problem mainly fall into three main categories: port-based, payload-based (e.g., deep packet inspection, short for DPI) and machine learning [6][14][19]. The classic port-based methods perform well for the applications with the specific port numbers (e.g., FTP traffic with port 21), but now these methods are seldom reliable since many applications adopt dynamic port allocation [23]. DPI analyzes the whole packet data and then identifies its network protocol and application. Essentially, this inspection can hardly do proper analysis to encrypted traffic, as the packet payloads, encrypted into a pseudo-random-like format, contain less constant features and indiscriminative patterns used for traffic classification. In addition, decrypting the encrypted traffic is not feasible, as it usually consumes significant computing resources. As for machine learning in traffic classification methods, its good performance is highly dependent on the excellent feature extraction and the classification algorithm selection, which are complicated and time-consuming. The advantages and drawbacks of machine learning approaches are analyzed in detail in Section II.

Given a network traffic flow, we intend to figure out whether it is encrypted or not, and then identify which application class (such as P2P and VoIP) it belongs to. The difficulty of this problem lies in how to select the discriminative features used for classification [27]. In most existing studies, features are designed and extracted by human experts [5][11-14][19]. This hand-designed feature selection consumes much time and cannot find out the exact features generally. There are only two studies [10][17] reported in the current literature that exploit the features automatically by using convolutional neural network on a packet level. However, using only one packet of a flow in these two studies is inadequate, since the other packets may contain different useful features (e.g., time sequence features between packets) for traffic classification. In this paper, we propose a new deep learning model for encrypted traffic classification. In this model, the effective Convolutional Neural Network (CNN) is adopted for packet feature extraction, and the powerful Long Short-Term Memory (LSTM) is used for time sequence feature extraction in a flow level. The accuracy and effectiveness of the proposed model are evaluated by the public dataset on Virtual Private Network (VPN) traffic. The main contributions of this paper are summarized as follows:

- Our model extracts features in both the packet level and the flow level automatically. The model dives into the time sequence features hidden in a flow, which has not been considered in the existing work but is effective in the problem of encrypted traffic classification. As a

978-1-5386-6614-2/18/$31.00 ©2018 IEEE
DOI 10.1109/HPCC/SmartCity/DSS.2018.00074

result, the proposed model can improve the accuracy of traffic classification by 5% in comparison with the state-of-art work, based on the public dataset on VPN traffic.

- Many existing studies classify the traffic by selecting the first *N* packets. This selection restricts the flexibility in the real-world environment. Differently, we tackle this problem by choosing any three consecutive packets in a flow.
- Many existing studies retain the basic flow information, including source and destination IP addresses, and source and destination port numbers. Their models are therefore prone to learning the classification evidence and can result in good performance. In contrast, in our experiments, we wipe out the basic flow information, and our model can still achieve excellent results.

The rest of this paper is constructed as follows: Section II presents the dataset and the related work about encrypted traffic classification using machine learning methods. Section III elaborates the proposed model, including the data pre-processing and model architecture. Section IV shows the experimental results based on the public dataset. Finally, we conclude the paper in Section V.

II. Background

A. Dataset

Even many studies are dedicated to the encrypted traffic classification, while few studies publish their datasets. In this paper, the public dataset published by the University of New Brunswick is selected in the evaluation of our proposed model: ISCX VPN-nonVPN traffic dataset [7]. The dataset contains 25GB raw traffic in the pcap format, which includes 14 network application classes, where 7 for regular traffic (such as Spotify and Facebook) and the rest for the corresponding traffic with VPN encrypted (such as VPN-Spotify and VPN-Facebook). We relabel the raw traffic into 12 classes, to be used in our experiments, according to the work [10]. The classes of the datasets applied to our model are listed in Table I in details. The dataset is split into the flow level according to the five tuples: {source IP, destination IP, source port, destination port, protocol}. We select the three consecutive packets at a random location in a flow, and randomly split the dataset into 80% training dataset and 20% testing dataset, respectively. More details about the dataset can be referred to [11].

TABLE I. Dataset labels

Type	Labels
Non-VPN	Email, Chat, Streaming, File, VoIP, P2P
VPN	VPN-Email, VPN-Chat , VPN-Streaming VPN-File, VPN-VoIP , VPN-P2P

B. Related work

We have briefly stated the drawbacks of the two main traffic classification methods: the port-based and the payload-based in the introduction. The merits and defects of important researches using machine learning methods for encrypted traffic classification will be elaborately discussed below. For clear comprehension, we categorize the related work into two classes: flow-feature-based and packet-byte-based.

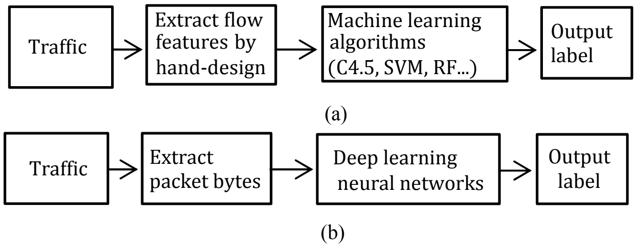

Traffic
Extract flow features by hand-design
Machine learning algorithms (C4.5, SVM, RF...)
Output label
(a)
Traffic
Extract packet bytes
Deep learning neural networks
Output label
(b)

Fig. 1. The two types of existing methods for encryted traffic classification: (a) flow-feature-based approach, (b) packet-byte-based approach.

1) Flow-feature-based approaches

We present the general procedure of flow-feature-based method in Fig. 1(a). The workflow goes as follows: the captured traffic traces are split into flows according to the five tuples. For each flow, this method extracts flow features by hand-design, such as duration per flow, flow bytes per second and the median of flow sizes. These features are fed into the classifier and trained with classification algorithms, such as C4.5 [12][19], SVM [13] and Random Forest [12]. Then the fine-trained classifier is ready to fulfill the classification task when a new traffic flow comes.

There are many studies reported in the literature for encrypted traffic classification by flow-feature-based methods. Sun et al. [5] proposed a hybrid method to address encrypted traffic classification. They first recognize TLS/SSL protocol by a static signature-based method. To identify the encrypted protocol, six flow features such as mean packet length and maximum and minimum packet length are adopted in training the model using Naive Bayes. Lashkari et al. [12] exploited 23 time-based handcraft features and utilized random forest algorithm on Weka [22], showing 83% accuracy in Tor traffic classification on the public Tor-nonTor dataset. Standing on their work, the authors in [13] improved the performance in Tor-nonTor traffic classification using artificial neural network and support vector machine. Although neural network is used, they still feed it with handcrafted flow features. Korczynski et al. [14] identified traffic flows in TLS/SSL sessions by constructing the Markov chain with stochastic fingerprint, but it needs to be updated periodically for the changing of fingerprint over time. In [19], the authors employed 39 packet header features and 22 statistical flow features for SSH and Skype identification. C4.5 and Adaboost are utilized to train these features. They achieve an average of 90% accuracy on six datasets. Alshammari [18] used C5.0 and genetic program to classify the encrypted VoIP traffic. In practice, the performance of this method relies on the handcraft feature engineering severely, which is time-consuming and resource-consuming.

2) Packet-byte-based approaches

As Fig. 1(b) shows, different from flow-feature-based approaches, packet-byte-based approaches evade the tedious

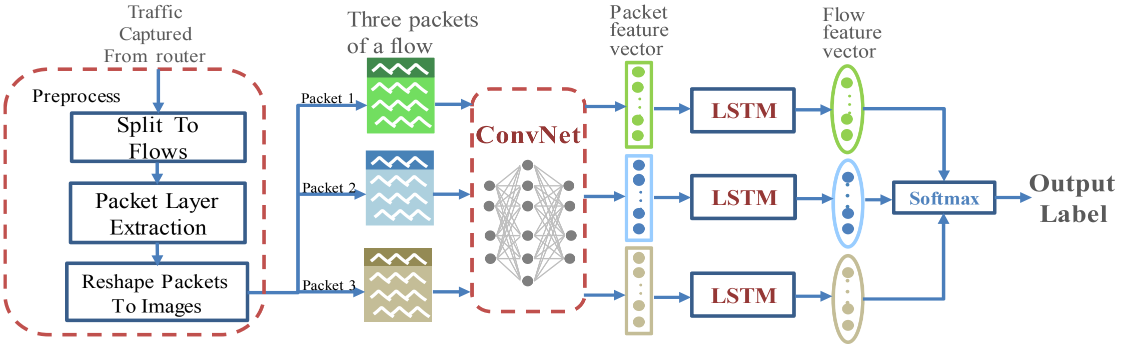


Fig. 2. The architecture of the proposed model. The traffic is first split into flows, and the three consecutive packets are selected at a random location in a flow. The three packet images of a flow, generated after the preprocess, are taken as the inputs of the convolutional neural network. The three packet feature vectors, the output of the convolutional part, are fed into the LSTM cells, which are followed by the softmax layer to output the probablity of each class.

work by utilizing the packet bytes directly. Taking the packet bytes as inputs, a deep learning neural network is trained for classification task. Using deep neural networks generally achieves the state that beats the flow-feature-based methods in effectiveness and efficiency.

To the best of our knowledge, only a few packet-byte-based methods address the need of traffic classification, two [10][17] of which attempted to classify encrypted traffic. Motivated by [16], the authors in [10] joined the first 784 bytes of each flow or session, which is trained with a 1D-convolutional neural network. They achieve the accuracy of 85.8% for encrypted traffic classification on public traffic datasets. Lotfollahi et al. [17] extracted the first 1500 bytes of only one packet and took them as the inputs of CNN. They train CNN and Stack Auto Encoder (SAE) to classify encrypted traffic. They both perform well in the classification problem, but they ignore the time sequence features hidden in a flow. In addition, the basic flow information (e.g., the five tuples) remains when training the model, which may result in poor scalability and performance in a different dataset.

III. The Proposed Model

Fig. 2 illustrates the workflow of our encrypted traffic classification, where the input to the system can be pcap files or actual traffic captured on the router. Before using the traffic as inputs into networks, we need to split the traffic into discrete units. Generally, there are two kinds of granularity in traffic classification: packet and flow. In this paper, we follow the flow granularity, which is more common in both the industry and the research community. For each flow, we extract the three consecutive packets. After the data pre-processing (Section III.A), we generate the three packet images. Then, we feed the convolutional neural network (Section III.B.1) with these three packet images. The outputs of the convolutional neural network are sent to the LSTM (Section III.B.2), which is followed by a softmax function and outputs the classification of this flow.

A. Data pre-processing

1) Flow splitting

We split the raw traffic into flows according to the five tuples. For any flow f_i in the flow set: $F=\{f_1, f_2, ...\}$, the number of packets f_i contains is n, i.e. $f_i=\{p_1^i, p_2^i, ..., p_n^i\}$. We select the three consecutive packets $P=\{p_k^i, p_{k+1}^i, p_{k+2}^i\}$ in the flow f_i at a random location. In other words, k is an integer number in the range $[1, n-2]$. Formally, we are training a function f or a classifier, aiming to identify the class a flow f_i belongs to (i.e., $\mathrm{f}:\{p_k^i, p_{k+1}^i, p_{k+2}^i\} \rightarrow class$).

2) Packet layer extraction

In our model, we make use of the packet bytes as the inputs of neural networks. Different packets convey bytes with different lengths, while the size of the inputs of neural networks is necessarily constant. Naturally, we need to constrain sampled packet bytes in a constant number. In our approach, the length of the payload byte sequence of any packet in P equals to 784. The data link header is wiped off, because it is stuffed by the two MAC addresses which contain useless features for traffic classification. We first collet the header of IP layer (we only consider the IPv4 traffic in this paper), which possesses 20 bytes. TCP/UDP layer is involved in our packet payload sequence. Additionally, we pad 12 zero bytes to the end of UDP header (8 bytes) for alignment with the TCP header in the same way as [17]. The remaining 744 bytes consist of the rest of the packet payload totally, truncating if the payload length exceeds 744, and padding zero if less.

To avoid the case where the network learns the classification evidence just according to the five tuples instead of the unique pattern of each class, we anonymize the flow information by making them be zeros, aiming to improving the scalability of our model.

3) Reshaping packets to images

For any packet in P, we have obtained a 784-byte sequence. As we know, one byte can be transformed into an integer in the range of [0, 255], so a byte can be viewed as a pixel. For convenience in computing, we reshape the 784-byte sequence to a 28*28 matrix. Correspondingly, the matrix is constructed as an image with 28-pixel width and 28-pixel height. At last, we normalize the scale to [0, 1] by dividing the bytes by 255. Until now, three packet images belonging to a flow are generated. These three 28*28 images will be fed into the neural network for training and testing.

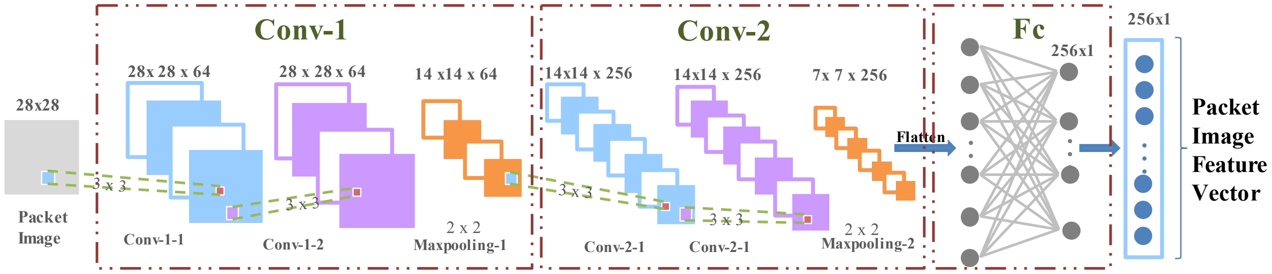


Fig. 3. The construct of the convolutional part in our model. The figure shows the process how the packet images go through the convolutional part. We take one packet image as an example. The rest two packet images in the same flow are trained similarly. The inputs of the CNN part are the packet image with the width of 28 and the height of 28. The outputs of the convolutional part is a packet image feature vector with the length of 256. Thus, the three packet image feature vectors can be generate

B. Architecture

Deep learning has been successfully employed in many fields [32], including computing vision [24], speech recognition [25] and Natural Language Processing (NLP) [23]. When it comes to traffic classification, we have drawn an analogy between them in Table II. As the work mentioned in Section III.A, we can use deep neural network to classify traffic like computing vision. In order to improve the utilization of the possibilities in neural networks, we combine the convolutional neural network and recurrent neural network in our model. Specially, the convolution neural network serves for digging the packet feature hidden inner a single packet image. It is worth to state that the three packets share the same convolutional neural network, which proves to be convenient and effective experimentally. The output of convolutional neural network is represented as three packet feature vectors with the same dimension, which are fed into the recurrent neural network for extracting the flow features of these three packets. The recurrent neural network is implemented by using the flexible LSTM cells. At last, we set a softmax layer to the end of the output of LSTM for the ultimate classification results.

TABLE II. THE COMPARISION OF NLP AND COMPUTING VISION WITH NETWORK TRAFFIC

Task	Unit Analogy			Field
Document Classification	Word	Sentence	Document	NLP
Video Classification	Pixel	Frame	Video	Computing vision
Flow Classification	Byte	Packet	Flow	Network traffic

1) ConvNet for packet image feature extraction

As Fig. 3 shows, the convolutional neural network part consists of two convolutional layers (Conv-1 & Conv-2) and a fully connect layer (Fc). As for the design of convolutional layer, we adopt the simple and effective VGG-net [28]. Conv-1 firstly twice filters the 28*28 image with 64 kernels, whose size is 3*3, corresponding to 3-grams bytes, with one stride. Relu [29] is employed as the activation function. After a 2*2 maxpooling layer, the output is sent to Conv-2, which has the same structure with Conv-1. Then we take the output of Conv-2 to the dense layer with 256 units. To prevent overfitting, the dropout layer with the 0.25 probability are added to the end of the Fc layer. We summarize the convolutional neural network part in Table III. We can get a packet image feature vector with 256 dimensions for a packet. When training or testing, the convolutional part transfers the three 28*28 packet images into three 256-dimension packet feature vectors.

TABLE III. THE DETAILS OF CONVOLUTIONAL PART

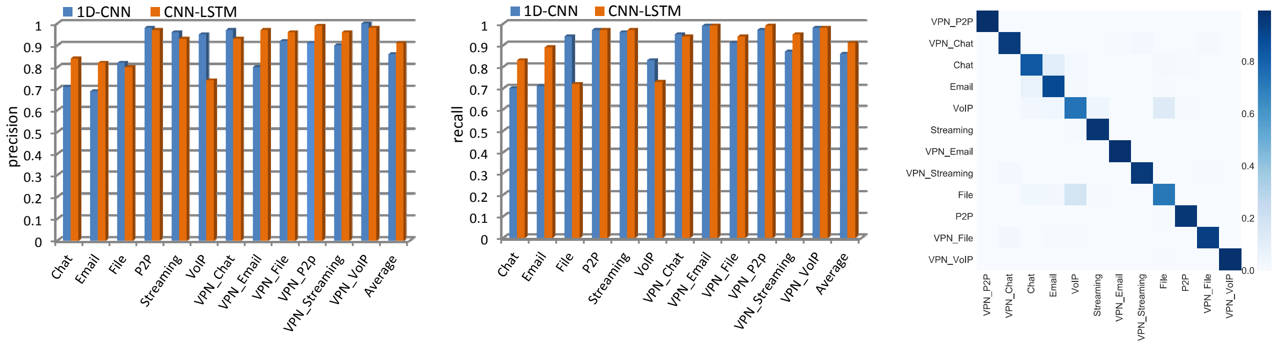
Layer	Name	Input	Filter	Stride	Output	Weight
Conv-1	Conv1-1	28*28	3*3*64	1	28*28*64	576
	Conv1-2	28*28*64	3*3*64	1	28*28*64	576
	Maxpooling-1	28*28*64	2*2	2	14*14*64	--
Conv-2	Conv1-1	14*14*64	3*3*128	1	14*14*128	1152
	Conv1-2	14*14*128	3*3*128	1	14*14*128	1152
	Maxpooling-2	14*14*64	2*2	2	7*7*128	--
Flat	Flatten	77128	-	-	6272	--
FC	Dense	6272	--	-	256	1605632

2) LSTM for flow time sequence feature extraction

Apart from the single packet features, the flow time sequence features, contained in the back and forth packets belonging to a flow between the source node and destination node, also attach important information to traffic classification. Recurrent neural networks have shown a considerable power for exploiting sequential tasks [9]. In this paper, we implement the recurrent layer by using the LSTM cells. All the states of the LSTM are initialized to be zero. We forward the three 256-dimension packet feature vectors to the LSTM cells with the time sequence $T = 3$. The number of the hidden units in LSTM is set to 256. Furthermore, we set the output probability with 0.8 in the LSTM to avoid over-fitting problem. Finally, the outputs of the recurrent layer are fed into a c -way softmax which produces a distribution over the c class labels. The variable c changes according to the number of the label classes in the dataset.

TABLE IV. THE DETAILS OF LSTM

Layer	Input	Output	Dropout	Weight
LSTM	256*3	256*3	0.8	262144
Softmax	256*3	c	--	256*3* c



(a) (b) (c)

Fig. 4 The results on the public dataset. (a) and (b) show the comparision results between the two models about the precision and recall respectively. (c) represents the confusion matrix of our model with rows normalized.

IV. Experiment Results

A. Experiment environment

In our experiments, we choose the VPN traffic to verify our model. The state-of-the-art study [10] (called 1D-CNN) is compared with our model in both efficiency and effectiveness by using the public dataset (ISCX VPN-nonVPN dataset). To implement our proposed model, the tensorflow framework [28] is employed on Ubuntu 16.04 OS with 52GB memory, equipped with a NVIDIA K80 GPU. The class labels are encoded as the one-hot vectors. T The training procedure is conducted by optimizing the sum objectives of categorical cross entropy and the L2 regularization [31] with the penalty of 0.05. We start up the Adam optimizer [30] with the initial learning rate at 1e-4, and a decay rate 0.9 in every epoch. The mini-batch size is set to be 128. Totally, the learning procedure stops after around 30 epochs.

B. Evaluation

To evaluate the performance of our model for traffic classification, we adopt the two common and widely-used metrics: precision and recall. These metrics are formulated as follows:

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

where TP and FP represent the number of items correctly and incorrectly labelled as belonging to the positive class, respectively. FN is the number of items incorrectly labelled as belonging to the negative class.

C. Experiment results and analysis

The overall comparison results between our model and 1D-CNN are exhibited in Figs. 4(a) and 4(b). Our model achieves the average precision of 0.91 and recall of 0.91, improving 5% compared with 1D-CNN on average. A first glance of the figure shapes the performance advantages of our model in contrast with 1D-CNN in different types of traffic. It hits the highest values in precision and recall for most of the types of traffic. The fact beneath the results suggests the effectiveness of the flow time sequence features by adding the function of LSTM layer in our model, and it further proves the combinational architecture of the convolutional layer and LSTM in encrypted traffic classification to be reasonable.

For further detailed analysis, we plot the heatmap of the confused matrix of our results in Fig. 5(c). As the figure shows, the six VPN traffic classes on the diagonal carry the deeper blue color, which shows the effective classification ability of our model on the VPN traffic. The six non-VPN traffic classes have the lighter blue, which suggests the weaker classification ability on the non-VPN traffic than the VPN traffic. The lightest blue color points to the class of file traffic. From this figure, we can see our model is prone to classifying the file to the VoIP. The main reason is that the captured traffic on the file transfer consists of the file transfer function on the Skype, which is also a typical VoIP software. It is worth to notice that, for one of the non-VPN traffic classes, the model only incorrectly classifies them into the other non-VPN traffic classes but not VPN traffic classes. It is the same with the VPN traffic classification where the model only incorrectly classifies it into the other VPN traffic classes but not non-VPN traffic classes. It therefore suggests we can almost achieve 0.99 accuracy and 0.99 recall when only classifying the traffic with the two classes of VPN and non-VPN traffic.

As the Table V shows, our model CNN-LSTM has the less number of weight and spends less time on training and testing. The results prove the efficiency of our proposed model.

TABLE V. The Time Consumption of The Models: 1D-CNN and CNN-LSTM

Model Name	Weights	Average Training time/epoch	Testing time
1D-CNN	5779k	70.186s	6.7126s
CNN-LSTM	3754k	63.165s	4.1603s

V. Conclusions

In this paper, we have proposed a novel model based on deep neural networks to classify the encrypted traffic automatically. The model has combined both the convolutional neural network and the recurrent neural network to extract the packet features of a single packet and time-sequence features hidden between packets in a flow, respectively. The LSTM cells have been adopted to implement the recurrent neural network. Experiment results have shown that our model outperforms the existing state-of-the-art model based on CNN, in terms of higher effectiveness and efficiency. We have concluded that the flow time-sequence features extracted by LSTM behave excellently on the encrypted traffic classification.

ACKNOWLEDGMENT

This work is supported by the National Key R&D Program of China (Grant No.:2017YFB0801801), the National Science and Technology Major Project of the Ministry of Science and Technology of China (Grant No.: 2017ZX03001019-003), and the Science and Technology Service Network Initiative (STS) Project of Chinese Academy of Science (Grant No.: Y7X0071105).

REFERENCES

[1] R. Dingledine, N. Mathewson, and P. Syverson. "Tor: the second-generation onion router," *Journal of the Franklin Institute*, vol. 239, no. 2, pp.135-139, 2004.

[2] M.-H. Maras, "Inside Darknet: The Takedown of Silk Road," *Criminal Justice Matters*, vol. 98, no. 1, pp. 22-23, 2014.

[3] Y. Wu, F. Hu, G. Min, and A. Zomaya (eds.), *Big Data and Computational Intelligence in Networking*, Taylor & Francis/CRC, ISBN: 9781498784863, 2017.

[4] C. Huang, G. Min, Y. Wu, Y. Ying, K. Pei, and Z. Xiang, "Time Series Anomaly Detection for Trustworthy Services in Cloud Computing Systems," *IEEE Transactions on Big Data*, 2017.

[5] L.G. Sun, Y. Xue, Y. Dong, and D. Wang "An Novel Hybrid Method for Effectively Classifying Encrypted Traffic," *IEEE Global Telecommunications Conference*, pp. 1-5, 2010.

[6] B. Yamansavascilar, et al. "Application identification via network traffic classification," *IEEE International Conference on Computing, Networking and Communications*, pp. 843-848, 2017.

[7] ISCX Vpn-nonVpn dataset, http://www.unb.ca/cic/datasets/vpn.html

[8] ISCX Tor-nonTor dataset, http://www.unb.ca/cic/datasets/tor.html

[9] J. Donahue, et al. "Long-term recurrent convolutional networks for visual recognition and description," *IEEE Computer Vision and Pattern Recognition*, pp. 677-691, 2015.

[10] W. Wei, M. Zhu, J. Wang, X. Zeng, and Z. Yang. "End-to-end encrypted traffic classification with one-dimensional convolution neural networks," *IEEE International Conference on Intelligence and Security Informatics*, pp. 43-48, 2017.

[11] H.A. Lashkari, et al. "Characterization of Encrypted and VPN Traffic Using Time-Related Features," *The International Conference on Information Systems Security and Privacy*, vol. 7, pp. 94–98, 2016.

[12] H.A. Lashkari, et al. "Characterization of Tor Traffic using Time based Features," *International Conference on Information Systems Security and Privacy*, pp. 253-262, 2017.

[13] E. Hodo, X. Bellekens, E. Iorkyase, A. Hamilton, C. Tachtatzis, and R. Atkinson, "Machine learning approach for detection of nontor traffic," in the *Proc. of the 12th International Conference on Availability, Reliability and Security*, New York, NY, USA: ACM, 2017

[14] M. Korczynski, and A. Duda. "Markov chain fingerprinting to classify encrypted traffic," *2014 Proceedings IEEE INFOCOM*, pp. 781-789, 2013.

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

[16] Z. Wang, "The application of deep learning on traffic identification," BlackHat USA , 2015.

[17] M. Lotfollahi, et al. "Deep Packet: A Novel Approach For Encrypted Traffic Classification Using Deep Learning," arXiv preprint arXiv:1709.02656, 2017.

[18] R. Alshammari, and A.N. Zincir-Heywood. "Identification of VoIP encrypted traffic using a machine learning approach," *Journal of King Saud University - Computer and Information Sciences*, vol. 27, no. 1, pp. 77-92, 2015.

[19] R. Alshammari, and A.N. Zincir-Heywood. "Can encrypted traffic be identified without port numbers, IP addresses and payload inspection?" *Computer Networks*, vol. 55, no. 6, pp. 1326-1350, 2011.

[20] L. Peng, H. Zhang, B. Yang, Y. Chen, and X. Zhou. "Early Stage Internet Traffic Identification Using Data Gravitation Based Classification. Dependable, Autonomic and Secure Computing," *Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress*, pp.504-511, 2016 .

[21] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian. "Traffic classification on the fly," *Acm Sigcomm Computer Communication Review*, vol. 36, no. 2, pp. 23-26, 2006.

[22] Weka3, https://www.cs.waikato.ac.nz/ml/weka/

[23] R. Collobert and J. Weston. "A unified architecture for natural language processing: deep neural networks with multitask learning," *International Conference on Machine Learning ACM*, pp. 160-167, 2008.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet classification with deep convolutional neural networks." *International Conference on Neural Information Processing Systems Curran Associates Inc*, pp. 1097-1105, 2012.

[25] L. Deng, et al. "Recent advances in deep learning for speech research at Microsoft," *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 8604-8608, 2013.

[26] TensorFlow, https://www.tensorflow.org

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

[28] K. Simonyan, A. Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition," *Computer Science*, 2014.

[29] V. Nair, and G. E. Hinton. "Rectified linear units improve restricted boltzmann machines," *International Conference on International Conference on Machine Learning*. Omnipress , pp.807-814, 2010.

[30] Kingma, Diederik P, and J. Ba. "Adam: A Method for Stochastic Optimization," *Computer Science*, 2014.

[31] Andrew Y. Ng. "Feature selection, L1 vs. L2 regularization, and rotational invariance," *Proceedings of the twenty-first international conference on Machine learning*, ACM, 2004.

[32] X. Yu, X. Wu, C. Luo, P. Ren. "Deep learning in remote sensing scene classification: a data augmentation enhanced convolutional neural network framework," *GIScience and Remote Sensing*, vol. 54, no. 5, pp. 741-758, 2017.