Malware traffic classification using convolutional neural network for representation learning

Conference Paper · January 2017

DOI: 10.1109/ICOIN.2017.7899588

5 authors, including:

Xuewen Zeng
Chinese Academy of Sciences
30 PUBLICATIONS **468** CITATIONS
SEE PROFILE

Xiaozhou Ye
Chinese Academy of Sciences
21 PUBLICATIONS **344** CITATIONS
SEE PROFILE

Malware Traffic Classification Using Convolutional Neural Network for Representation Learning

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

Xuewen Zeng, Xiaozhou Ye, Yiqiang Sheng
National Network New Media Engineering Research Center,
Institute of Acoustics, Chinese Academy of Sciences
Beijing, China
{zengxw, yexz, shengyq}@dsp.ac.cn

***Abstract*—Traffic classification is the first step for network anomaly detection or network based intrusion detection system and plays an important role in network security domain. In this paper we first presented a new taxonomy of traffic classification from an artificial intelligence perspective, and then proposed a malware traffic classification method using convolutional neural network by taking traffic data as images. This method needed no hand-designed features but directly took raw traffic as input data of classifier. To the best of our knowledge this interesting attempt is the first time of applying representation learning approach to malware traffic classification using raw traffic data. We determined that the best type of traffic representation is session with all layers through eight experiments. The method is validated in two scenarios including three types of classifiers and the experiment results show that our proposed method can satisfy the accuracy requirement of practical application.**

Keywords—traffic classification; convolutional neural network; representation learning; network anomaly detection; intrusion detection system

I. INTRODUCTION

Traffic classification is the task of associating network traffic with the generating application, which has been a task of crucial importance in the network management and especially network security domains. In network security domain, traffic classification represents in fact the first step for activities such as anomaly detection for the identification of malicious use of network resources [1].

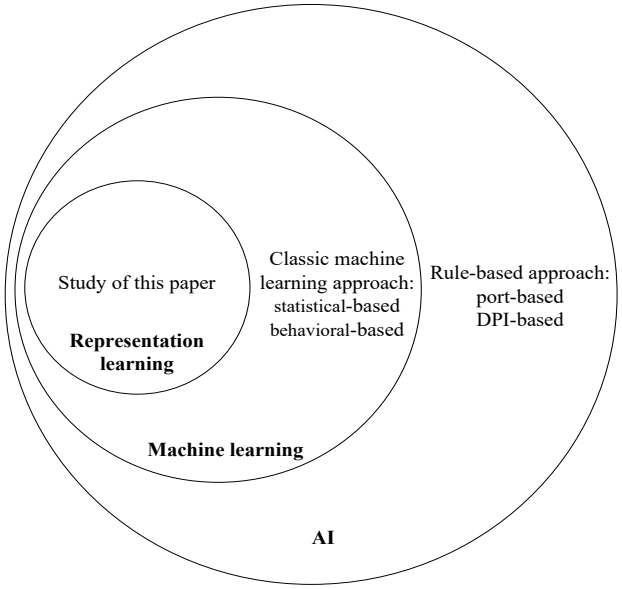

Fig. 1. Traffic classification taxonomy in an AI perspective

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

There are four main traffic classification methods [1]: port-based, deep packets inspection (DPI)-based, statistical-based, and behavioral-based. From the perspective of artificial intelligence (AI) development [2], port-based and DPI-based methods are rule-based approaches, which perform traffic classification by matching predefined hard-coded rules. Statistical-based and behavioral-based methods are classic machine learning approaches, which classify traffic by extracting patterns from empirical data using a set of selective features. Although classic machine learning approach solves many issues that rule-based approach cannot solve, such as encrypted traffic classification and high computational cost, it faces a new challenge of designing proper features, and many recent studies focus on this problem [3].

Representation learning is a new rapidly developing machine learning approach in recent years that automatically learning features from raw data and to a certain extent has solved the problem of hand-designing features [4]. Especially, the deep learning method which is a typical approach of representation learning has achieved very good performance in many domains including image classification and speech recognition [5] [6]. The main goal of this paper is to attempt to apply representation learning approach to malware traffic classification domain and demonstrate its effectiveness. Figure 1 illustrates the traffic classification taxonomy in an AI perspective. Figure 2 shows different work flows of these approaches, and shaded boxes indicate components that are able to learn from data [2].

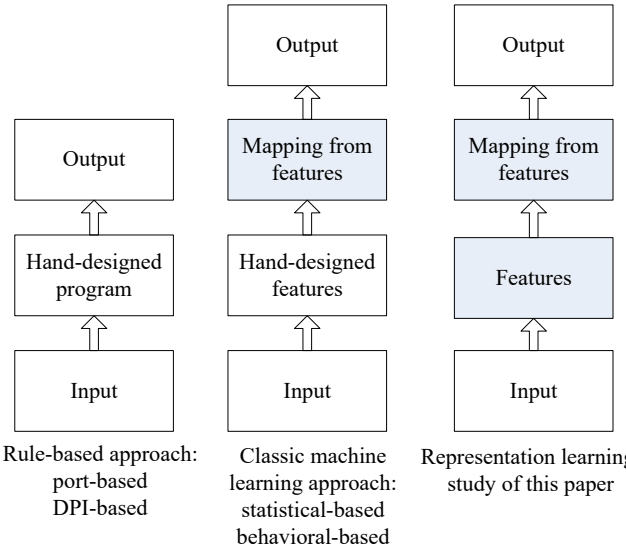

Fig. 2. Work flows of traffic classification in an AI perspective

The technique this paper used is convolutional neural network (CNN) which is currently one of the most popular representation learning methods. We didn't extract features

from traffic but took raw traffic data as images, and then used CNN to perform image classification that is the most common task to CNN [7], finally the goal of malware traffic classification was achieved. To the best of our knowledge, this interesting attempt is the first application of representation learning approach to malware traffic classification domain using raw traffic data. Because of the difference between the continuity of traffic data and the discreteness of image data, multiple traffic representation types were studied and the best type was found by experiments. To demonstrate the scalability of our proposed method, we conduct experiments in two scenarios using three types of classifiers, and the final average accuracy is 99.41%, meeting the practical application standard. Besides, a traffic dataset USTC-TFC2016 was created and a data-preprocessing toolkit USTC-TK2016 was developed. Both of them and our training and test source codes in our work will all be published on GitHub to interested researchers.

The remainder of this paper is organized as follows. Section II describes related work and introduces motivation of our work. Section III describes the methodology of the proposed method using CNN. Section IV presents experiment results and analysis. Section V describes limitation of our current method and future work. Section VI provides concluding remarks.

II. RELATED WORK

Rule-based traffic classification approach is relatively mature in industry, and related work mainly focused on how to exactly extract mapping rules and improve performance. Finsterbusch et al. [8] summarized current main DPI-based traffic classification methods. Classic machine learning approach of traffic classification attracts a lot of research in academia, and related work mainly focused on how to choose a better feature set. Dhote et al. [3] provided a survey on feature selection techniques of internet traffic classification.

There are a few of research about representation learning based traffic classification now. Gao et al. [9] proposed a malware traffic classification method using deep belief networks. Javaid et al. [10] proposed a malware traffic identification method using sparse auto encoder. Those researches both used deep learning technique and applied to design of network-based intrusion detection systems (NIDS). But there is a same problem in their study that they both used hand designed flow feature dataset as input data, (KDD CUP1999 & NSL-KDD respectively [11]). The greatest advantage of representation learning is being able to learning features directly from raw data, and the successful application of deep learning in image classification domain and speech recognition domain has sufficiently proved it. Unfortunately, the aforementioned two studies just gave up that advantage. Wang [12] proposed a stacked auto encoder (SAE) based network protocol identification method using raw traffic data, and achieved high accuracy. Traffic classification and protocol identification are very similar tasks. Therefore, it is reasonable to presume that representation learning method can achieve good performance in malware traffic classification task.

In this paper, we are motivated by those studies to perform malware traffic classification task using raw traffic data, and the representation method we used is CNN which is better than SAE in image classification task.

III. METHODOLOGY

A. DataSet

As Dainotti et al. [13] pointed out, lack of a variety of sharable traces dataset as test data is the most obvious obstacle to progress on traffic classification. A lot of research about malware traffic classification used self-collected traffic or private traffic of security companies, damaging the credibility of their results. Because classic machine learning approaches focus on feature selection techniques, many current public traffic datasets are flow features datasets other than raw traffic datasets, e.g. famous KDD CUP1999 and NSL-KDD provide pre-defined 41 features in their dataset [11], and there are many similar datasets, such as [14]. These datasets don't meet our requirement to raw traffic. In datasets providing raw traffic, there are few containing sufficient both normal and malware traffic data, such as [15].

To solve those problems, a dataset USTC-TFC2016 is created. The dataset consists of two parts, as shown in Table I and II. Part I contains ten types of malware traffic from public website which were collected from real network environment by CTU researcher from 2011 to 2015 [16]. For some traffic with too big size, only a part of them was used. For some traffic with too small size, we merged them which are generated by the same application. Part II contains ten types of normal traffic which were collected using IXIA BPS [17], which is a kind of professional network traffic simulation equipment. The information on simulation method details can be found at their product website. To reflect more kinds of traffic as possible, ten kinds of traffic contains eight classes of common application. The size of USTC-TFC2016 dataset is 3.71GB, and the format is pcap. It will be published on GitHub for interested researchers.

TABLE I. USTC-TFC2016 PART I (MALWARE TRAFFIC)

Name	CTU num	Binary MD5	process
Cridex	108-1	25b8631afeea279ac00b2da70fffe18a	original
Geodo	119-2	306573e52008779a0801a25fafb18101	part
Htbot	110-1	e515267ba19417974a63b51e4f7dd9e9	original
Miuref	127-1	a41d395286deb113e17bd3f4b69ec182	original
Neris	42,43	bf08e6b02e00d2bc6dd493e93e69872f	merged
Nsis-ay	53	eaf85db9898d3c9101fd5fcfa4ac80e4	original
Shifu	142-1	b9bc3f1b2aace824482c10ffa422f78b	part
Tinba	150-1	e9718e38e35ca31c6bc0281cb4ecfae8	part
Virut	54	85f9a5247afbe51e64794193f1dd72eb	original
Zeus	116-2	8df6603d7cbc2fd5862b14377582d46	original

TABLE II. USTC-TFC2016 PART II (NORMAL TRAFFIC)

Name	Class	Name	Class
BitTorrent	P2P	Outlook	Email/WebMail
Facetime	Voice/Video	Skype	Chat/IM
FTP	Data Transfer	SMB	Data Transfer
Gmail	Email/WebMail	Weibo	Social NetWork
MySQL	Database	WorldOfWarcraft	Game

B. Network Traffic Representation

Machine learning based traffic Classification approach need to split continuous traffic to discrete units based on certain granularity at first. On the other hand, different OSI or TCP/IP layers can be chosen in every packet. The following part introduces how to select traffic granularity and packet layers in our approach.

1) Traffic granularity

Network traffic split granularity include: TCP connection, flow, session, service, and host [13]. Different split granularity results in different traffic units. Our method used flow and session which were also used by most researchers. A flow is defined as all packets that has the same 5-tuple, i.e. source IP, source port, destination IP, destination port and transport-level protocol. A session is defined as bidirectional flows, including both directions of traffic. The formal description is described as follows:

- Raw traffic: All packets are defined as a set $P = \{p^1, ..., p^{|P|}\}$, and every packet is defined as $p^i = (x^i, b^i, t^i), i = 1, 2, ..., |P|$. The 1st element x^i stands for a 5-tuple, and the 2nd element stands for the size of packet $b^i \in [0, \infty)$ in bytes, and the last element stands for the start time of transmission $t^i \in [0, \infty)$ in seconds.
- Flow: A set of raw traffic P can be divided into multiply subsets. All packets in a subset are arranged in time order, i.e. $\{p^1 = (x^1, b^1, t^1), ..., p^n = (x^n, b^n, t^n)\}$, $t^1 < t^2 < ... < t^n$. A subset is defined as a flow $f = (x, b, d_t, t)$. The 1st element is the same 5-tuple, i.e. $x = x_1 = ... = x_n$. The 2nd element is the sum of size of all packets in a flow. The 3rd element is the flow duration $d_t = t^n - t^1$. The last element is the start time of transmission of the first packet. The whole raw traffic can be then converted into flows $F = \{f^1, ..., f^n\}$.
- Session: A session includes both directions of flows, i.e. the source and destination IP / port are interchangeable.

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. We can give an intuitive explanation to this choice. In general, the front part of a flow or session is usually connection data and a few of content data and should be best of reflecting the intrinsic characteristics of a flow or session. This choice is very similar with [18, 19] which researched malware traffic identification with classic machine learning approach. Besides, because only first hundreds bytes are used, this method can be much more lightweight than many rule-based methods.

2) Packet layers

From the perspective of packet layers analysis, intuitively the intrinsic characteristics should be reflected in application layer of TCP/IP model, i.e. layer 7 of OSI model. For example, STMP protocol represents email traffic and HTTP protocol represents browser traffic. Based on this assumption, Wang [12] only select layer 7 and he call it TCP session payload. On the other hand, other layers' data should also contain some traffic feature information. For example, port information in transport layer can identify most applications using standard port number. Sometimes the flag information can identify network attack e.g. SYN attack and RST attack. So two types of packet layers choice were used: all layers (All) and only layer 7 (L7). It should be noted that the IP and MAC information in a session or flow may damage the feature extraction procedure. To eliminate that negative impact, we must remove that information through randomization, usually called traffic anonymization or sanitization [20].

In summary, there are four types of traffic representation: Flow + All, Flow + L7, Session + All, Session + L7. Their performances were tested using two types of traffic dataset introduced in part A, and finally the best representation type was determined. So there were eight experiments totally in this study and the results and analysis are shown in section IV.

C. Data Preprocessing

Data preprocessing is the procedure of converting raw traffic data (pcap format) to CNN input data (idx format). It includes four steps: traffic split, traffic clear, image generation and IDX conversion, and a toolkit USTC-TL2016 was developed accordingly. Figure 3 shows the whole process of data preprocessing.

Step 1 (Traffic Split). This step split a continuous raw traffic to multiply discrete traffic units. Input data format is pcap. If representation type is Flow + All or Session + All, output data format is pcap. If representation type is Flow + L7 or Session + L7, output data format is bin

Step 2 (Traffic Clear). This step first performs traffic anonymization / sanitization, which randomizes MAC address and IP address in in data link layer and IP layer respectively. That is optional, for example, when all traffic is from one same network, the MAC and IP may be no longer the distinguishing information, and in this situation we need not perform it. The second action of traffic clear is files clean. Some packets have no application layer, so the result bin files are empty. Some packets generate identical files when they have the same content, and duplicated data can result in bias when training CNN. Those empty and duplicated files need to be removed. The data format in this step has no change.

Step 3 (Image Generation). This step first trims all files to uniform length. If file size is larger than 784 bytes, it is trimmed to 784 bytes. If file size is shorter than 784 bytes, the 0x00 is added in the end to complement it to 784 bytes. The result files with the same size are then converted to gray images. Each byte of original file represents a pixel, e.g. 0x00 is black and 0xff is white. In fact, this conversion is optional, and we can simply convert files to IDX files directly. This conversion is only to show researchers the visual images, so they can be analyzed in a visual way.

Step 4 (IDX conversion). This step converts images to IDX format files. An IDX file contains all pixels and statistics information of a set of images. IDX format is a common file format in machine learning field [21].

The USTC-TFC2016 traffic dataset was processed using the USTC-TK2016 toolkit, and generated 752,040 records in total. Table III shows the results. Because sessions include bidirectional flows, the number of sessions is usually fewer than flows.

TABLE III SESSIONS & FLOWS COUNT RESULT

Dataset	Representation	Count Range	Count Total
Malware	Flow+ALL	6000~17178	134563
	Flow+L7	4569~13968	76716
	Session+ALL	6000~8629	71008
	Session+L7	4569~7592	63120
	Total	———	**406633**
Benign	Flow+ALL	10051~17008	138145
	Flow+L7	8908~16391	126665
	Session+ALL	5134~9634	71692
	Session+L7	4952~9576	70131
	Total	———	**345407**
Total	———		**752040**

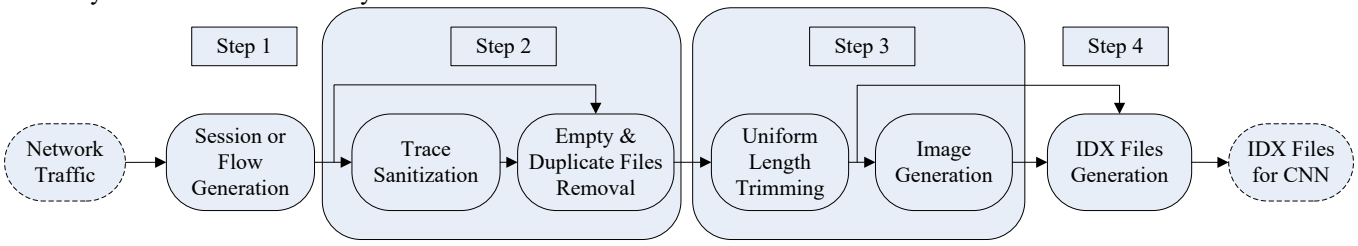

Fig. 3. Data Preprocess Procedure

D. Visualization Analysis

The images generated after the step 3 of data preprocess procedure are analyzed in this part. The size of each grey image is 784 (28*28) bytes. The visualization results of Session + All are shown in Figure 4 and 5. The result of other three types of representation is generally similar with them.

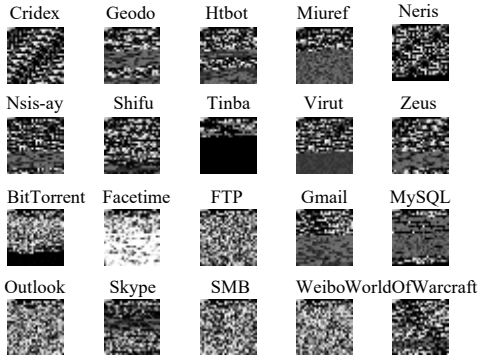

Fig. 4. Visualization of All Classes of Traffic

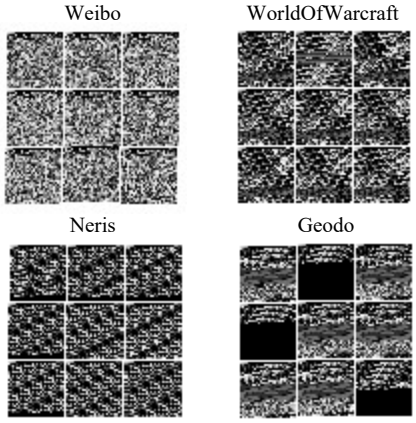

Fig. 5. Consistency in the same traffic class

Figure 4 shows the visualization result of all classes. It is obvious that they are easy to distinguish from each other. Only a few of images are very similar, e.g. FTP and SMB. Figure 5 shows the consistency in the same traffic class. Nine randomly selected images in one class and four randomly selected classes are shown. It is also obvious that the texture of images in Weibo, WOW and Neris are very similar. Only Geodo images can be divided into two subclasses, but even the texture of images in respective subclasses are still very similar. Other sixteen classes have generally similar consistency.

Through visualization analysis, we concluded that different classes of traffic have obvious discrimination degree and each class of traffic has high consistency, so it was reasonable to presume that our approach could achieve good performance.

E. CNN Architecture

The size of preprocessed dataset and the size of each image are both very similar to classic MNIST dataset [21], so a CNN architecture similar to LeNet-5 [22] was used but with much more channels in each convolution layer.

CNN first reads traffic image of size of 28*28*1 from IDX files. Those image pixels are normalized to [0, 1] from [0, 255]. The first convolution layer C1 performs a convolution operation with 32 kernels of size of 5*5. The results of C1 layer are 32 feature maps of size of 28*28. There is a 2*2 max pooling operation in P1 layer followed C1 layer and the results are 32 feature maps of size of 14*14. The kernel size of the second convolution layer C2 is also 5*5 but has 64 channels. The results are 64 feature maps of size of 14*14. After the second 2*2 max pooling layer P2, 64 feature maps of size of 7*7 are generated. The last two layers are fully connection, the result sizes are 1024 and 10 respectively. A softmax function is used to output the probability of each class. Besides, dropout is used to alleviate over-fitting. That CNN architecture is used in all three types of classifiers to be introduced in part F.

F. Scalability Study

Two application scenarios including three types of CNN classifiers were used to validate the scalability of the proposed method: binary classifier, 10-class classifier, 20-class classifier. Figure 6 shows the experiment setup of two scenarios. We used the best traffic representation type determined in part B.

Scenario A (Binary and 10-class classifiers). In real application of traffic classification, the most common requirement is only the malware traffic identification, which is also the main goal of NIDS. If necessary, more fine traffic classification can be performed to identify each class of malware and normal traffic. In this scenario, a binary classification was firstly performed to identify malware or normal, and then performed two 10-class classifications respectively to identify each class of traffic.

Scenario B (10-class and 20-class classifiers). In some applications, we need to classify all types of traffic at one time, which demands relatively high performance to classifier. A 20-class classification using all classes of traffic was performed.

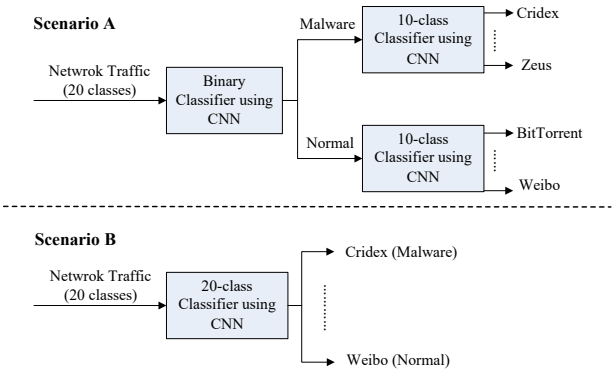

Fig. 6. Two Scenarios including three classifiers

IV. EVALUATION

A. Experiment Setup

The TensorFlow [23] is used as experiment software framework which runs on Ubuntu 14.04 64bit OS. Server is DELL R720 with 16 cores CPU and 16GB memory. An Nvidia Tesla K40m GPU is used as accelerator. One tenth 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 0.001, training time is about 40 epochs.

B. Evaluation Metrics

Four evaluation metrics were used: accuracy (A), precision (P), recall (R), f1 value (F1). Accuracy was used to evaluate the overall performance of a classifier. Precision, recall and f1 value were used to evaluate performance of every class of traffic.

$$A = \frac{TP + TN}{TP + FP + FN + TN}, \quad P = \frac{TP}{TP + FP}$$

$$R = \frac{TP}{TP + FN}, \quad F_1 = \frac{2PR}{P + R}$$

Where 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.

C. Representation Experiment Results And Analysis

Eight experiments were carried out to determine the best traffic representation type. Figure 7 shows the accuracy of four representation types on malware and normal traffic dataset. It can be found from four comparisons that the accuracy of traffic class with all layers is always higher than with only L7 layer. Except that the accuracies of session and flow with L7 in normal traffic are about equal, other three comparisons all show that the accuracy of traffic class using session is always higher than using flow. Figure 8 shows the precision, recall and f1 value of Nsis-ay which is one of 20 classes of traffic. We can see the same pattern that Figure 7 shows. In all twelve comparisons, except that the recall of session using all layers is slightly lower (0.24%) than flow using all layers, other eleven comparisons show the following pattern: the precision, recall, f1 value of traffic class with all layers were all higher than with only L7 layer, and the precision, recall, f1 value of traffic class using session were all higher than using flow. Not only that, the same pattern can be found in other 19 classes of traffic results. In summary, All is better than L7 and Session is better than Flow.

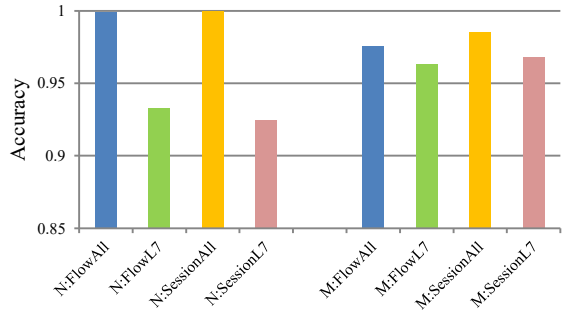

Fig. 7. Accuracy of four representations with two classes of traffic

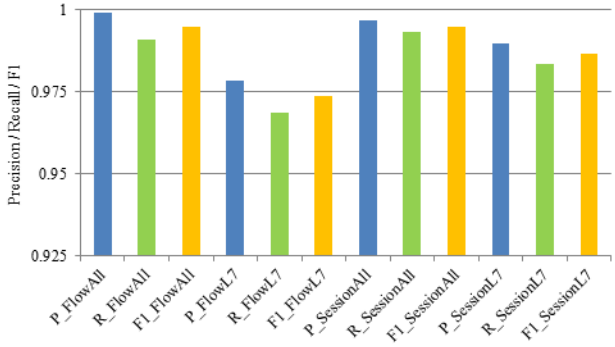

Fig. 8. Precision, recall, f1 value of Nsis-ay traffic

An intuitive explanation can be given for that pattern. Because session contains bidirectional flows so contains more interaction information than unidirectional flow. Because all layers representation contains more layers than L7 layer, especially includes port and flag information, so it can represent more key information, which proved our assumption in part B of section III. It should be noted that L7 layer preprocessing result records of many class of traffic are much fewer than all layers preprocessing result records, e.g. the L7 layer records of CTU 107-1 traffic is only 8 but all layers records are 16386. When generating the same amount of training data, much more traffic data are needed using L7 than all layers representation. So we can see that the all layer representation is more flexible than L7 layer representation.

In summary, it can be found that the best type of traffic representation is Session + All. Only that type of traffic representation was used to carry out scalability experiments.

D. Scalability Experiment Results And Analysis

Table IV shows the overall accuracy of three types of classifiers in two scenarios using Session + All traffic representation. Table V and VI show the precision, recall and f1 value of each class of traffic. Because the accuracy of binary classifier is 100%, it's no need to show the precision, recall and f1 value of binary classifier.

TABLE IV Accuracy of three classifiers (%)

Metric	Average	Binary Classifier	10-class Classifier		20-class Classifier
			Normal	Malware	
Accuracy	99.41	100	99.94	98.52	99.17

TABLE V Precision, recall and f1 value of 20-classifiers (%)

DATA	PR	RC	F1	DATA	PR	RC	F1
Cridex	100	100	100	BitTrt	100	100	100
Geodo	100	99.9	99.9	Facetime	100	100	100
Htbot	99.8	100	99.9	FTP	100	100	100
Miuref	100	100	100	Gmail	98.4	99.3	98.8
Neris	96.3	92.9	94.6	MySQL	100	100	100
Nsis-ay	99.8	99.0	99.4	Outlook	99.2	98.1	98.7
Shifu	99.9	99.9	99.9	Skype	99.8	100	99.9
Tinba	100	100	100	SMB	100	100	100
Virut	90.7	95.6	93.1	Weibo	100	100	100
Zeus	100	100	100	Wow	100	99.9	99.9

TABLE VI Precision, recall and f1 value of 10-classifiers (%)

DATA	PR	RC	F1	DATA	PR	RC	F1
Cridex	100	100	100	BitTrt	100	100	100
Geodo	100	100	100	Facetime	100	100	100
Htbot	99.8	99.8	99.8	FTP	100	100	100
Miuref	100	100	100	Gmail	99.9	99.7	99.8
Neris	97.1	91.1	94.0	MySQL	100	100	100
Nsis-ay	99.7	99.3	99.5	Outlook	99.6	99.9	99.7
Shifu	99.9	99.8	99.8	Skype	100	100	100
Tinba	99.9	100	99.9	SMB	100	100	100
Virut	89.0	96.5	92.6	Weibo	100	100	100
Zeus	100	100	100	Wow	100	100	100

Table IV shows that the accuracy of four classifiers are very high, even the lowest accuracy already achieves 98.52%. Table V shows that the precision, recall and f1 value of Neris and Virus traffic is a bit lower (90% ~ 96%), but the other 18 classes of traffic all achieve very high precision, recall and f1 value (higher than 99%). Table VI shows similar patterns and the metrics of Neris and Virus traffic is a bit lower (89% ~ 97%), but the other 18 classes of traffic all achieve very high metrics (higher than 99%). The reason why Neris and Virus appear slightly lower metrics maybe their special characteristic and need to be explored further.

In summary, the average accuracy of all three classifiers is already 99.41% and meets accuracy demands of practical use.

E. Comparison

It is not easy to conduct a fair comparison among various malware traffic classification methods due to differences between traffic datasets, software setup, and experimental environment. On the other hand, all three types of classifiers of our method have achieved a pretty high accuracy. Therefore, instead of doing a performance comparison between our method and other existing methods, we compare them in terms of some significant characteristics. Table VII shows a general comparison between our method and rule based Snort which is a well-known NIDS, classic machine learning based Celik [18], and aforementioned Gao [9] and Javaid [10] which both use hand-designed flow features.

TABLE VII COMPARISON OF DIFFERENT METHODS

Characteristics	Our's	Snort	Celik	Gao	Javaid
Early stage detection	Yes	No	Yes	No	No
Low false alarm rate	Yes	No	Yes	Yes	Yes
Protocol independent	Yes	No	Yes	Yes	Yes
Features auto extraction	Yes	No	No	Yes	Yes
Raw Traffic data input	Yes	Yes	No	No	No

Our method has ability of early stage malware traffic detection due to using only the front hundreds of bytes of each session. Snort need to matching traffic fingerprint whose position is uncertain. Gao and Javaid need to extract features from the whole traffic flow. So they both have no early stage detection characteristic. The experimental results show that our method has low false alarm rate, and Snort has relatively high false alarm rate due to some problems such as difficulty of precisely extracting malware traffic footprints. Our method is protocol independent due to using image classification approach, and Snort need hand designed matching rules for each protocol. Our method can auto extract features and Celik has no such ability which is based on classic machine learning approach such as SVM. Finally, our method directly uses raw traffic dataset, by contrast, Gao and Javaid both use hand designed flow features dataset.

V. LIMITATION AND FUTURE WORK

There are three limitations and related future work about our work. Firstly, the main goal of this paper is to prove the efficiency of malware traffic classification using representation learning approach, so we didn't study the CNN parameters tunings. The traffic size and classes number are certainly not fixed in real applications, so in machine learning terms, the generalization capability of our approach need to be further validated. Secondly, traditional traffic classification methods using classic machine learning approach use many temporal (time series) features and has proved their high efficiency [24]. Our approach in fact only uses spatial features of traffic, and totally ignored temporal features. We will research how to add these temporal features in our representation learning approach in future work, e.g. using recurrent neural network (RNN). Lastly, our work in this paper only performs known malware traffic classification. The capability of identify unknown malware traffic is also very important to a NIDS [25], how to add the capability of unknown malware traffic to our approach needs to be further studied in future work.

VI. CONCLUSION

On the basis of systemic summarization of traditional malware traffic classification methods, a new malware traffic classification method using representation learning approach was proposed. Our method need not to hand-design traffic features beforehand. The raw traffic data is directly the input data of traffic classifier, and the classifier can learn features automatically. CNN was used as our representation learning technique in experiments. The USTC-TRC2016 traffic dataset was created and a data preprocessing toolkit named USTC-TK2016 was developed. Based on the dataset and the toolkit, through analyzing eight experiments results, the best traffic representation type was found. The experiment results using three types of classifiers in two scenarios show that our method achieved accuracy demands of practical use, and has high scalability. In future work, we plan to further research the approach proposed in this paper to improve the capability of malware traffic identification.

REFERENCES

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

[2] I. Goodfellow, Y. Bengio, and A. Courville, Deep learning, Book in preparation for MIT Press, 2016.

[3] Y. Dhote, S. Agrawal. "A Survey on Feature Selection Techniques for Internet Traffic Classification". in Computational Intelligence and Communication Networks, Jabalpur, 2015, pp. 1375-1380.

[4] Y. Bengio, A. Courville and P. Vincent, "Representation learning: A review and new perspectives", IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, pp. 1798-1828, Aug. 2013.

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

[6] Z. Qingqing, L. Yong, W. Zhichao, P. Jielin and Y. Yonghong, "The Application of Convolutional Neural Network in Speech Recognition", Microcomputer Applications, vol. 3, pp. 39-42, June. 2014.

[7] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy and B. Shuai, "Recent Advances in Convolutional Neural Networks", arXiv preprint arXiv:1512.07108, 2015.

[8] M. Finsterbusch, C. Richter, E. Rocha, J. A. Muller and K. Hanssgen, "A Survey of Payload-Based Traffic Classification Approaches", Communications Surveys & Tutorials IEEE, vol. 16, no. 2, pp. 1135-1156, 2014.

[9] N Gao, L Gao and 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.

[10] 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.

[11] M. Tavallaee, E. Bagheri, W. Lu and A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set", Proc. 2009 IEEE Int. Conf. Comput. Intell. Security Defense Appl., pp. 53-58.

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

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

[14] G. Creech and J. Hu, "Generation of a new ids test dataset: Time to retire the kdd collection", Wireless Communications and Networking Conference (WCNC) 2013 IEEE, pp. 4487-4492, 2013.

[15] F. Haddadi and A. Nur Zincir-Heywood, "Data confirmation for botnet traffic analysis," in Proc. 7th Int. Symp. FPS, to be published.

[16] CTU University, The Stratosphere IPS Project Dataset, https://stratosphereips.org/category/dataset.html, 2016.

[17] Ixia Corporation, Ixia Breakpoint Overview and Specifications, https://www.ixiacom.com/products/breakingpoint, 2016.

[18] Z. B. Celik, R. J. Walls, P. McDaniel and A. Swami, "Malware traffic detection using tamper resistant features," Military Communications Conference, MILCOM 2015 - 2015 IEEE, Tampa, FL, 2015, pp. 330-335.

[19] W. Li, "Efficient Application Identification and the Temporal and Spatial Stability of Classification Schema", Computer Networks, vol. 53, pp. 790-809, Apr. 2009.

[20] D. Koukis, S. Antonatos, D. Antoniades, E. P. Markatos and P. Trimintzios, "A Generic Anonymization Framework for Network Traffic," 2006 IEEE International Conference on Communications, Istanbul, 2006, pp. 2302-2309.

[21] IDX File Format Specifications, Behaviour and Example, http://www.fon.hum.uva.nl/praat/manual/IDX_file_format.html, 2016.

[22] Y. A. LeCun, L. D. Jackel, L. Bottou, A. Brunot, C. Cortes, J. S. Denker, et al., "Learning algorithms for classification: A comparison on handwritten digit recognition", Neural Networks, pp. 261-276, 1995.

[23] 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.

[24] A. W. Moore, D. Zuev, and M. Crogan. Discriminators for use in flow-based classification. Technical Report RR-05-13, Department of Computer Science, Queen Mary, University of London, September 2005.

[25] MH. Bhuyan, DK. Bhattacharyya and JK. Kalita, "Network Anomaly Detection: Methods, Systems and Tools," in IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 303-336, First Quarter 2014.