Trojan Traffic Detection Based on Meta-learning

Zijian Jia[1,2], Yepeng Yao[1,2], Qiuyun Wang[1], Xuren Wang[1,3], Baoxu Liu[1,2], and Zhengwei Jiang[1,2]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{jiazijian,yaoyepeng,wangqiuyun,liubaoxu,jiangzhengwei}@iie.ac.cn

[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

[3] College of Information Engineering, Capital Normal University, Beijing, China
{wangxuren}@cnu.edu.cn

Abstract. At present, Trojan traffic detection technology based on machine learning generally needs a large number of traffic samples as the training set. In the real network environment, in the face of Zero-Day attack and Trojan variant technology, we may only get a small number of traffic samples in a short time, which can not meet the training requirements of the model. To solve this problem, this paper proposes a method of Trojan traffic detection using meta-learning for the first time, which mainly includes the embedded part and the relation part. In the embedding part, we design a neural network combining ResNet and BiLSTM to transform the original traffic into eigenvectors and allocate the meta tasks of each round of training in the form of a C-way K-shot. In the relation part, we design a relationship network improved by dynamic routing algorithm to calculate the relationship score between samples and categories in the meta-task. The model can learn the ability to calculate the difference between different types of samples on multiple meta-tasks. The model can use a small number of samples to complete training and classify quickly according to prior knowledge. In few-shot, our method has better results in Trojan traffic classification than the traditional deep learning method.

Keywords: Network security · Trojan traffic detection · Deep learning · Meta learning.

1 Introduction

In recent years, the number of network attacks is gradually increasing with the development of the Internet. Trojan is one of the means of attack. Attackers usually implant the Trojan virus into the victim's host to control the victim's host through this program. Because there must be communication behavior between the attacker and the victim host, detecting the traffic generated in it has become an extremely effective method to detect the Trojan virus.

At present, many researchers use machine learning method for traffic detection.They extract many features from the original traffic and use machine learning methods to detect and classify malicious traffic[2, 3]. But the feature extraction is very subjective. With the continuous development of deep learning, more and more people begin to build various neural network models to detect Trojan traffic behavior and content. For example, the researcher transformed the payload of traffic into a gray image and used CNN to train the recognition model. Or the payload is directly regarded as semantic information for numerical calculation. They use RNN and other networks to train the recognition model directly.All these works show that the use of deep learning in traffic classification can achieve better results.

Compared with traditional machine learning, deep learning has significant advantages because it does not need to extract features manually.It mostly avoid the influence of human subjective factors on the model. But the premise of this model training is to have a large number of data as training samples so that the model can obtain enough knowledge from the iteration and adjust the parameters to achieve a better classification effect. According to the current real cyberspace situation, such a premise is difficult to achieve. Trojan attacks often change their behavior, leading to the problem of slow sample capture and a large number of sample data for training cannot be obtained quickly.

In this paper, we propose a meta-learning method based on the metric to solve this problem. We transform the original network traffic into feature vectors. An improved relational network is constructed to calculate the difference between different vectors. Our model obtains prior knowledge through a large number of meta-task and learns to calculate the differences between different samples so that the Trojan traffic detection can be implemented when the dataset is imbalance. The main contributions of this method are the following three aspects:

- We propose a network combining ResNet and BiLSTM as the embedded part of meta-learning. On a large number of dataset, this network's result is better than that of a general neural network, which can better learn and extract the features of the original traffic.
- We use a dynamic routing algorithm to replace the superposition part of vectors in a traditional relational network. This algorithm can make the vector fusion simultaneously retain more information, such as the direction of the vector. As the representative vector of category in meta task is more accurate.
- Based on the CTU13 dataset, we build an unbalanced dataset and compare the standard deep learning model's classification effect and the proposed meta-learning model. The results show that our model is better than the general deep learning model when the dataset is imbalance.

2 The Description of Problems

The general deep learning to find a mapping relationship between sample x and label y such that $f(x) = y$. If the dataset is imbalance, it is easy to overfit the sample recognition. To solve this problem, we propose a meta-learning method based on metric for Trojan traffic detection. This method contains a task set $T = \{T_i, i \in N^*\}$ for training. This setting can make the model through the training of n tasks to use the prior knowledge obtained from the existing tasks to complete training quickly. The representation of meta-learning architecture is shown in figure 1. It contains Meta-training set D_{train} and Meta-testing set D_{test}. Each item in the two sets is a small meta-task T which containing the Support Set and the Query Set.

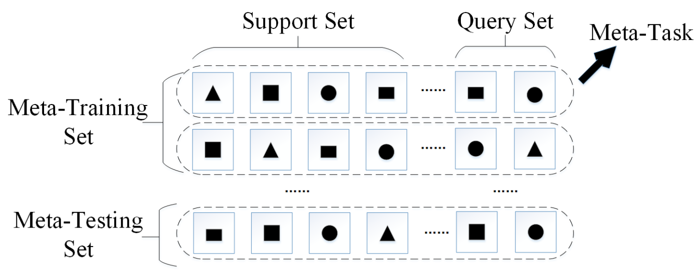

Fig. 1. The representation of Meta learning

The task allocation can be summarized as a C-way K-shot problem. Each task randomly selects C classes from the dataset, and K samples from each classes. The support set contains different Trojan sample traffic classes and the query set includes the same kind of Trojan sample traffic as the support. The model is trained by meta-task like figure 1.When the training is completed, the model can calculate the score between the unknown sample and each category in new task to achieve samples' classification.

3 Trojan traffic detection based on meta-learning model

3.1 Transform Trojan traffic into feature vector

At present, many researchers use deep learning to solve the problem of network traffic detection. No matter what the network structure design of deep learning is, they all need a fixed input mode. In this chapter, a neural network based on ResNet-LSTM is proposed to extract the original traffic feature vectors as the embedded part of the meta-learning framework. The primary extraction process is shown in Figure 2.

Data preprocessing of pcap package The model extracts the two-way TCP flow from the packet as the sample. After extracting the data stream, we need to filter some of the information.

The traffic in the dataset is captured by a trojan virus running in the virtual machine environment. The MAC address and other header's information in the

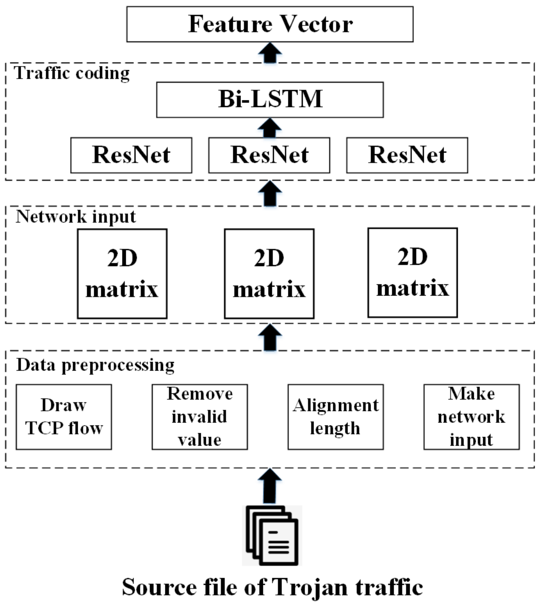

Fig. 2. The Extraction of Feature Vector

packet header will interfere with the model learning. So we should remove this information. According to Rezaei's research[4], the characteristics of the first few packets can represent the whole data flow. In the experiment, we focus on the initial stage of the whole TCP flow. According to the experimental results, the first three to five packets of the data stream are needed. Data cleaning is needed after extracting the data stream. The header information such as five tuples in each packet is cleared and the payload part in TCP flow is directly extracted. This part extracts 784 bytes. If the length is not enough, we fill zero at the end. Then a 28x28 two-dimensional matrix is formed as the input of the embedded part of the model.

Make feature vector by ResNet-LSTM Most of researchers build 3 to 5 layers of CNN to detect traffic and don't explore more deep network. A large number of experiments show that the network's recognition efficiency decreases when the depth of the general CNN network increases gradually. He K et al. proposed the ResNet model, which can deepen the network level and make the whole network have better learning ability[1]. In this paper, this model is combined with bidirectional LSTM to extract the original traffic characteristics in space and time. The ResNet model designed in this experiment is shown in table 1.

The Conv1 layer transform the sample into one hundred and twenty-eight 28x28 characteristic matrices. Then there are three blocks of ResNet. This net-

Table 1. The structure of ResNet

Network Name	Network Output	Network Structure
Conv1	$128 \times 28 \times 28$	$1 \times 1, 128$
Conv2_Res	$128 \times 14 \times 14$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$
Conv3_Res	$256 \times 7 \times 7$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$
Conv4_Res	$512 \times 4 \times 4$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$
Ave_pool	$512 \times 1 \times 1$	4×4
Full_connected	512	512

work is made up of CNN with a jump layer. When the depth of the network deepens, it is equivalent to mapping the current network to the next layer. If there is an identity mapping between the network and the next layer, the model will degenerate into an external network, and there is no degradation problem. This is what the residual network looks like, as shown in Figure 3.

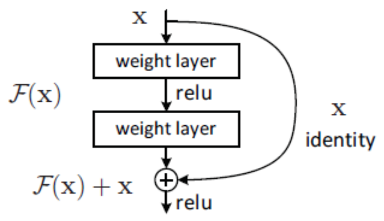

Fig. 3. Neuron structure of residual network[1]

The three residual blocks contain different convolution layers. In Table 1, network structure represents the shape of every network layer. At last, the generated characteristic matrix is input into a fully connected layer as the next layer's input to bidirectional LSTM.

A convolutional neural network extracts the spatial characteristics of the content in the traffic. The traffic is continuous in terms of time and the payload in the communication about Trojan traffic has semantic properties. In the paper, the bidirectional LSTM model is used to encode the features of the ResNet. It mainly processes the two unidirectional LSTM networks' output and obtains the feature vector used to represent the sample, which provides a unified input for the meta-learning model.

3.2 The Structure of Meta-learning model

As we all know, neural network iterates and optimizes by learning many labeled dataset. So when the dataset is imbalance, the model will be overfitted. In this model, the neural network can learn how to calculate the difference between a sample and the class in different tasks. In this way, the model has the ability of generalization. Even if the samples of a single class are unbalanced, the model can also help classify the samples through the knowledge of comparison. The overall framework is shown in figure 4.

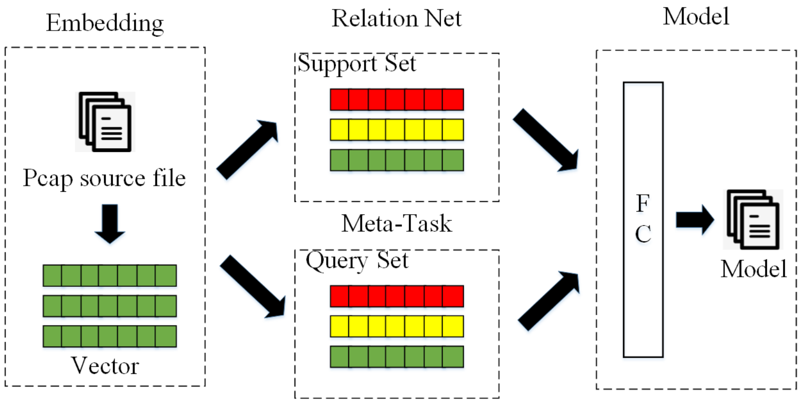

Fig. 4. The Structure of Meta-learning

The whole model consists of three parts. The first part is the embedding part. This part aims to extract the original Trojan traffic samples, transform them into computable feature vectors, and then divide the model into meta tasks. Each meta task contains a support set and a query set. The second part is the relation network based on metric. Facing each meta task, the model will fuse the samples contained in all categories in the support set so that all classes have their representative vectors. The vector and all samples in the query set are input into the full join layer to calculate the relationship score. The network is optimized iteratively through the score. The third part is the verification part. The samples to be tested are input into the trained model in the form of tasks, and the classification results are obtained according to the scores between samples and classes.

Task generation in meta-learning The training set and test set contain meta-tasks in this model, including samples selected from different classes. The specific algorithm is shown in algorithm 1.

Algorithm 1 Generate One Meta-Task For Training

Input: Labelset $L = \{1, 2, ..., n\}$,Dataset$D = \{(X_1, L_1), ..., (X_n, L_n)\}$,$L_i \in L$,$X_i$ is the i kind of sample,$X_i = \{x_1, ..., x_n\}$,select K_s samples from Support Set,select K_q samples from Query Set

Output: Meta learning task$T = \{Support, Query, L_{Query}\}$,$S$ is Support Set,Q is Query Set,L_Q is Labsels of Query Set.

Require: $RandomSample(A, K)$ Indicates that k samples are randomly selected from the A set,$Remove(A, B)$,Indicates that the B sample is removed from the A set

```
1: procedure GenerateTask($D, K_s, K_q$)
2:     GenerateSet:
3:     for $i = 1$ to $n$ do                      ▷ Traverse all categories in the dataset
4:         $S_{L_i} \leftarrow RandomSample(X_i, K_s)$
5:         $Other \leftarrow Remove(X_i, S_{L_i})$
6:         $Q_{L_i} \leftarrow RandomSample(Other, K_q)$
7:     end for
8:     GenerateTask:
9:     $Support \leftarrow \sum_1^n S_i$                 ▷ Splice the selected samples
10:    $Query \leftarrow \sum_1^n Q_i$
11:    $L_{Query} \leftarrow \sum_1^n L_i xlen(Q_i)$
12:    return $T \leftarrow \{Support, Query, L_{Query}\}$
13: end procedure
```

Meta-Learning models based on metric After meta-task assignment, each task will contain different Trojan class feature vectors and each class has K samples. When K is greater than 1, Sung et al. used an embedding module to sum each class's elements step by step in early study[14]. This embedding module consists of a convolutional neural network. The combined class level mapping feature is combined with the mapping feature in the query set to calculate the relationship score. The number of samples in the support set is greater than 1 in our experiment. We use the dynamic routing algorithm from the capsule network to fuse the vectors.

Hinton et al. first proposed this algorithm[17]. Using this method is that when the traditional convolutional neural network does the dimension mapping from the bottom to the top[18, 19], the relative relations such as direction and space information in the vector cannot be learned in the pooling layer. In our experiment, we need to carry out high-dimensional mapping on multiple vectors of the same kind and different vectors will also have deviations in the direction. Such information can not be transmitted upward using simple superposition. So the dynamic routing method is adopted, which encapsulates the space state of the underlying vector and passes it to the next neuron to complete the expression of different Trojan class vectors. The algorithm is shown in algorithm 2.

This kind of neuron calculates vector. Its essence is to update the initial weight value of b_{ij} through vector iteration. This step's update is equivalent to using the high-level output vector and the low-level input vector for dot product operation update. When the two vectors' directions are the same, the point product operation will increase the weight update value and vice versa. The

Algorithm 2 Dynamic routing algorithm

Input: Vector of samples v_{ij} in Support Set S,Initialization vector $b_{ij} = 0$, Number of Route iterations r
Output: The Vector represent a Class V_k in Support Set S

```
 1: procedure DYNAMIC($v_{ij}, b_{ij}, r$)
 2:     for r iterations do
 3:         for $i$ in layer $l$ do
 4:             $c_i \leftarrow softmax(b_i)$
 5:         end for
 6:         for $j$ in layer $(l+1)$ do
 7:             $s_j \leftarrow \sum_{j=1}^{n} c_{ij} v_{ij}$
 8:             $v_j \leftarrow squash(s_j)$                ▷ Squeezing function
 9:         end for
10:         for $i$ in layer $l$ and $j$ in layer $(l+1)$ do
11:             $b_{ij} \leftarrow b_{ij} + v_i * v_j$
12:         end for
13:     end for
14:     return $v_j$
15: end procedure
```

algorithm repeats the iteration process until r times. Hinton also pointed out that a large number of iterations will lead to overfitting[1]. They recommended using three iterations. So we use the same super parameters for training.

After fusion, we can obtain the representative eigenvectors of different classes of Trojan traffic in meta-task. They are used as the support set to calculating the relation score with other classes of samples in the query set. Two vectors are input into two fully connected layers, calculate relation score by $softmax$ and obtain a classification result. The score is also used to calculate the loss value of the task. As the whole training is multitasking, the model uses the MSE as the loss value and the formula is shown in (1).

$$L = \sum_{i=1}^{C} \sum_{j=1}^{K} (r_{ij} - 1 * (y_j == y_i))^2 \tag{1}$$

In formula (1), C stands for the selected C classes, K stands for K samples, the final output of the model is the relation score between classes and samples. The label of classes judged by the score. The *Adam* and *SGD* optimizers are compared on the optimizer. And *Adam* is selected as the optimization algorithm of the model according to the experimental results. The purpose of this meta-learning model after optimization is to get the metric function G_Φ, as shown in (2).

$$r_{ij} = g_\phi(C(f_\phi(x_i), f_\phi(x_i))) \qquad i = 1, ..., C \tag{2}$$

Through the neural network's nonlinear optimization process, a function g_ϕ which can express the metric between vector is obtained. Each task generated in training is to generalize this form of metric calculation so that the network can learn the ability of comparison. When new tasks appear, the network can calculate the differences between samples and different classes appropriately through the comparison function to achieve classification.

4 Evaluation

4.1 Dataset

To test the performance of the model, we use the CTU13 dataset. It's from the statosphere lab project of Czech Polytechnic University. It's often used in the industry which contains the malicious traffic generated by many classic Trojans. Ten Trojans are selected for experiment and analysis as shown in table 2.

Table 2. The Scale of Dataset

Types of Trojan horse	The Size of Original Data(M)	Samples(TCP)
Andromeda	620	29343
Emotet	500	23444
Geodo	2560	22938
Locky	372	55207
Sathurbot	1510	68820
Tinba	1100	34450
Trojan.Rasftuby	1210	41447
Variant	1460	32756
Yakes	1350	41725
Zeus	782	64161

Table 2 shows the classes of Trojans that build the experimental dataset. The original pcap packet samples are cleaned and each bidirectional TCP flow is extracted as a sample. The samples that do not contain any payload information are removed. Due to network traffic characteristics, each byte is between 0 and 255, just in the gray values range. Each sample is saved in the form of a gray image as the input part of the meta-learning model.

4.2 The Results of Experiments

Firstly, we evaluate the embedded part of the model. In the experiment, we construct different neural networks to learn the original TCP stream's content and check the classification effect. We have constructed four kinds of neural networks, including CNN, BiLSTM, ResNet and ResLSTM. The detection results of these four kinds of neural networks are shown in Table 3. This experiment uses all the data sets, the data sets are balanced, each class contains a large number of samples, mainly used to verify the ability of the embedded part of the network to identify the original traffic.

Table 3. The Results of Embedding

	Precision	Recall	F1-Score
CNN	94.7%	94.5%	94.6%
BiLSTM	96.4%	98.9%	97.6%
ResNet	99.0%	96.3%	97.6%
ResLSTM	99.9%	99.9%	99.9%

The experimental results are the average of the recognition accuracy and recall rate of different Trojan traffic in the dataset. It can be seen from the experimental results that the method of combining ResNet and BiLSTM can obtain a better recognition effect in a large number of the dataset. Therefore, this method can better extract features from the original traffic. So in the experiment, we use the model proposed in the third section to encode the original traffic and transform the original traffic into a computable numerical vector in space.

After determining the embedded model of meta-learning, we need to verify the whole meta-learning model's detection ability in the case of few-shot. This small sample setting is that when the unknown Trojan traffic appears, the model only uses a small amount of data collected quickly for training and then detects and classifies the subsequent traffic. To verify this case, we first set the traffic generated by any Trojan horse in the dataset as unknown attack traffic and reduce the data volume to 0.25% of the original. The reserved dataset is used as a training set and test set. It together with other Trojan traffic to form an unbalanced dataset. The remaining data of this category is used as the validation set after model training.

The data in Table 4 shows the comparison of different deep learning models. There are five models, including CNN, BiLSTM, ResLSTM, the original relational network and our the model of this paper. The first three are standard deep learning models, and the last two are meta-learning models. We use the method proposed by Wang et al[5]. ResLSTM uses the model we put forward in the embedded layer and input the model's output directly into the softmax function for classification operation. In the case of few-shot, although the recall of this model is not as good as the CNN and BiLSTM, the precision becomes higher, and the F1 is higher than them. The latter two models adopt the measurement-

Table 4. Experimental results of few-shot

	Precision	Recall	F1-Score
CNN	75.4%	75.9%	75.6%
BiLSTM	74.1%	76.8%	75.4%
ResLSTM	78.8%	75.3%	77.0%
Relation Net	81.7%	76.1%	78.8%
Method in Paper	85.7%	82.9%	84.2%

based meta-learning model. Relation Net uses the relational network proposed by Sung. In the meta-learning model, we set super parameters in advance when assigning meta tasks. Each time we select all ten categories, the number of samples K in each type is set to 10 , half of which is divided into a support set, and half is placed as the query set for training.

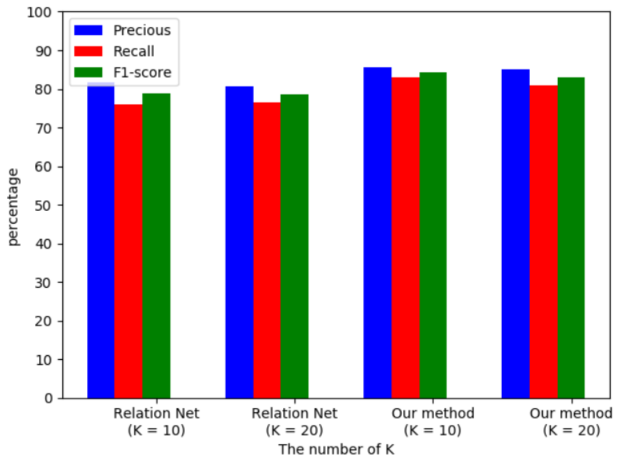

Fig. 5. The results of experiments about different K

At the same time, we also tested the influence of the sample size of K on the results. In the C-way K-shot problem, the number of samples selected for each class is generally less than 20. In the experiment's task allocation phase, we test the impact of K = 10 and 20 on the experimental results. This allocation method can ensure that the number of samples in the support set and query set is 5 and 10. The results are shown in Figure 5. The selection of K has little effect on the results. This may be due to the significant difference between different traffic. The model can be used as the representative vector of each class only using a small number of samples for fusion.

5 Related work

5.1 Trojan traffic detection

It is challenging to prevent Trojan's attack, so we need to detect according to the communication behavior after intrusion. Li et al. analyzed Trojan's actual network behavior, extracted features from the TCP handshake process and detected them[20]. D. Jiang et al. extracted the number and length of packets in the early communication stage Trojan[21]. They established a machine learning model for detection. This kind of research extract features from Trojan communication and builds models.

5.2 Traffic detection based on deep-learning

In recent years, researchers gradually apply deep learning methods to traffic detection. Wang et al. proposed a variety of malicious traffic detection methods using deep learning. They used 1D-CNN, 2D-CNN and other methods to model the payload information in malicious traffic[5–7]. Ren Hwang et al. used LSTM to transform TCP flow into word vector training and established a traffic classification model[8]. Xue Liu et al. used a bidirectional GRU network and attention mechanism to classify encrypted traffic[9]. In the aspect of traffic detection and classification, the deep learning method can get good results.

The research about detecting malicious traffic on imbalance dataset generally use the data enhancement method[16] and meta-learning method is rarely used. Xu et al. first proposed a network intrusion detection system based on a meta-learning framework[15]. They use metric-based learning models to construct a binary classifier, distinguishing malicious traffic from normal traffic. We have carried on the exploration in this aspect and proposed the method in this paper.

5.3 Research status of meta-learning

At present, meta-learning network mainly include three categories: model-based network, optimization-based network and metric-based network. The model-based network method is to quickly update parameters on a small number of samples through the model structure's design[10, 11]. The optimization-based network completes the task of small sample classification by adjusting the optimization method[12].

The metric-based network calculates the distance between the samples in the training set and the test set. Koch et al. proposed the Siamese Network[13]. Different pairs of samples are constructed by combination and input into the network for training. The distance between two samples is used to judge whether they belong to the same class. Sung et al. proposed the relational network, which considers that the measurement method needs to be modeled[14]. They built a neural network(CNN) to learn how to calculate distance between two samples.

6 Conclusion

We propose a Trojan traffic detection method based on meta-learning, which can classify Trojan traffic. In the classification task, the model preprocesses the original pcap packet, uses ResLSTM network as the embedded part to encode the feature vector of the sample, and then calculates the association score between the sample and each category, and judges whether the sample belongs to the class through the association score. To evaluate the method's effectiveness, we select different Trojan traffic types from CTU13 and construct a data set with the balanced and unbalanced sample number. The experimental results show that the imbalanced data set makes the traditional deep learning model overfit, and the detection result is lower than the sample balanced data set. The meta-learning method can achieve better detection results on many verification sets after training on unbalanced data sets.

In future research, we will continue to explore the use of other meta-learning methods to detect Trojan traffic. And we will analyze the non-standard encrypted Trojan traffic to study the model's detection ability for this traffic.

Acknowledgement This research is supported by the National Key Research and Development Program of China (Grant No. 2018YFC0824801). It is also partially supported by Key Laboratory of Network Assessment Technology, Chinese Academy of Sciences and Beijing Key Laboratory of Network Security and Protection Technology.

References

1. He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.
2. MOORE A W, ZUEV D. Discriminators for use in flow-based classification[R]. Technical report, Intel Research, Cambridge, 2005.
3. Feizollah A, Anuar N B, Salleh R, et al. Comparative study of k-means and mini batch k-means clustering algorithms in android malware detection using network traffic analysis[C]//2014 International Symposium on Biometrics and Security Technologies (ISBAST). IEEE, 2014: 193-197.
4. Rezaei S, Liu X. Deep learning for encrypted traffic classification: An overview[J]. IEEE communications magazine, 2019, 57(5): 76-81.
5. Wang W, Zhu M, Zeng X, et al. Malware traffic classification using convolutional neural network for representation learning[C]//2017 International Conference on Information Networking (ICOIN). IEEE, 2017: 712-717.

6. Wang W, Zhu M, Wang J, et al. End-to-end encrypted traffic classification with one-dimensional convolution neural networks[C]//2017 IEEE International Conference on Intelligence and Security Informatics (ISI). IEEE, 2017: 43-48.
7. Wang W, Sheng Y, Wang J, et al. HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection[J]. IEEE Access, 2017, 6: 1792-1806.
8. Hwang R H, Peng M C, Nguyen V L, et al. An LSTM-Based Deep Learning Approach for Classifying Malicious Traffic at the Packet Level[J]. Applied Sciences, 2019, 9(16): 3414.
9. Liu X , You J , Wu Y , et al. Attention-Based Bidirectional GRU Networks for Efficient HTTPS Traffic Classification[J]. Information Sciences, 2020, 541.
10. Santoro A, Bartunov S, Botvinick M, et al. One-shot learning with memory-augmented neural networks[J]. arXiv preprint arXiv:1605.06065, 2016.
11. Munkhdalai T, Yu H. Meta networks[C].Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, 2017: 2554-2563.
12. Ravi S, Larochelle H. Optimization as a model for few-shot learning[J]. International Conference on Learning Representations. 2017.
13. Koch G, Zemel R, Salakhutdinov R. Siamese neural networks for one-shot image recognition[C]//ICML deep learning workshop. 2015, 2.
14. Sung F, Yang Y, Zhang L, et al. Learning to compare: Relation network for few-shot learning[C].Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 1199-1208.
15. Xu C , Shen J , Du X . A Method of Few-Shot Network Intrusion Detection Based on Meta-Learning Framework[J]. IEEE Transactions on Information Forensics and Security, 2020, PP(99):1-1.
16. Zhenyan L , Yifei Z , Pengfei Z , et al. An Imbalanced Malicious Domains Detection Method Based on Passive DNS Traffic Analysis[J]. Security and Communication Networks, 2018, 2018:1-7.
17. Sabour S, Frosst N, Hinton G E. Dynamic routing between capsules[C].Advances in neural information processing systems. 2017: 3856-3866.
18. Mandal B, Ghosh S, Sarkhel R, et al. Using dynamic routing to extract intermediate features for developing scalable capsule networks[C].2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP). IEEE, 2019: 1-6.
19. Lin A, Li J, Ma Z. On learning and learned data representation by capsule networks[J]. IEEE Access, 2019, 7: 50808-50822.
20. S. Li, X. Yun, Y. Zhang, J. Xiao and Y. Wang, ”A General Framework of Trojan Communication Detection Based on Network Traces”, IEEE Seventh International Conference on Networking Architecture and Storage, pp. 49-58, 2012.
21. D. Jiang and K. Omote, ”An Approach to Detect Remote Access Trojan in the Early Stage of Communication,” 2015 IEEE 29th International Conference on Advanced Information Networking and Applications, Gwangiu, 2015, pp. 706-713, doi: 10.1109/AINA.2015.257.