PROCEEDINGS OF SCIENCE

PoS(ISGC2022)027

A Weakly-supervised Method for Encrypted Malicious Traffic Detection

Junyi Liu,[a,b,*] **Zhenyu Li,**[a,c] **Jiarong Wang** ✉,[a] **Tian Yan,**[a] **Dehai An,**[a] **Caiqiu Zhou**[a] **and Gang Chen**[a]

[a] *Computing Center, Institute of High Energy Physics, Chinese Academy of Sciences, Beijing 100049, P.R.China*

[b] *School of Nuclear Science and Technology, University of Chinese Academy of Sciences, Beijing 100049, P.R.China*

[c] *School of Cyber Science and Engineering, Zhengzhou University, Zhengzhou 450001, P.R.China*

E-mail: wangjr@ihep.ac.cn

[*] Speaker

In order to defend against complex threats and attacks in cyberspace, IHEPSOC has been developed and deployed at the Institute of High Energy Physics Chinese Academy of Sciences (IHEP), which is considered as an integrated security operation platform to ensure a secure state of the network and scientific researches in IHEP. Nowadays the integration of the state-of-the-art cyber-attack detection algorithms for IHEPSOC has become an important task for enhancing the attack discovery capability of IHEPSOC. Meanwhile, malicious traffic detection comes to be increasingly challenging. With the extensive application of data encryption techniques to protect communication security and privacy, malicious software could also hide their attack information, making most of malicious traffic identification methods such as port-based methods and DPI (Deep Packet Inspection)-based methods ineffective.

Machine learning based detection methods have been proposed to address the encrypted malicious traffic detection problem, which usually involve constructing statistical features of internet traffic flow and training classification models for detecting encrypted malicious traffic. There are some drawbacks in the above-mentioned detection methods. First of all, feature selection is a time-consuming procedure that depends on expert experience. Secondly, most traffic classification schemes employ supervised learning methods, and the acquisition of large fine-grained labeled datasets is a tedious task. In this paper, we propose a weakly-supervised method for encrypted malicious traffic detection, which combines the generative adversarial network (GAN) and the multiple instance learning detector to achieve the fine-grained classification of encrypted traffic with a small number of coarse-grained labeled samples and a large number of unlabeled samples. Thus, we could focus on the accuracy of the malware detector instead of spending efforts on dataset annotation with the weakly-supervised learning approach. To start with, we convert the traffic data into single-channel grayscale images in the proposed method, and then input them into the GAN, so that the original traffic features can be learned without manual effort. Therefore, the improved semi-supervised learning generative adversarial network, which is based upon convolution neural network (CNN) architecture, generates more synthetic samples for data augmentation and addresses the insufficiency of labeled samples. In addition, a multi-instance detector with an attention mechanism is used to identify encrypted malicious traffic from coarse-grained labeled data. We validate the proposed approach on two public datasets. Compared with other malicious traffic detection methods, the experimental results show that our proposed framework can effectively perform fine-grained detection of encrypted malicious traffic.

International Symposium on Grids & Clouds 2022 (ISGC 2022)
21 - 25 March, 2022
*Online, Academia Sinica Computing Centre (ASGC), Taipei, Taiwan****

1. Introduction

In recent years, with the rapid development of the Internet, especially the mobile Internet and the Internet of Things, IT devices have proliferated, and network traffic has become massive and complicated. How to effectively analyze and utilize network traffic in the explosive growth of data has become an important research direction. At the same time, security breaches, ransomware, APT attacks, and other network security threats emerge in an endless stream, causing serious harm to the normal work and life of ordinary users and the entire Internet environment. The global and distributed nature of high-energy physics computing makes it more vulnerable to threats from different geographical areas and different attack vectors. To counter these threats, cooperation within the high-energy physics community is required, while a proper Security Operations Centre (SOC) is essential. In conjunction with the Institute of High Energy Physics of the Chinese Academy of Sciences, the IHEPSOC[1] was developed, integrating several tools and models working in tandem to enable functions such as threat intelligence sharing and network intrusion detection.

For more accurate malicious traffic detection, traffic classification is required. Traffic classification is usually the first step in network anomaly detection and plays an important role in the field of network security. It is of great significance for recording and managing network operating conditions, defending against cyber attacks, and ensuring the quality of service.[4] Depending on the granularity of network traffic, traffic can be classified according to different levels such as network protocols, network traffic types, and network applications. Network protocol classifies traffic into different protocols[22], such as SSL, DNS, and SMTP. Network traffic type classifies traffic into traffic types [22] such as P2P and VoIP. Network application classifies traffic into specific applications [22] such as Wechat, Gmail, Youtube, etc. The finer the granularity of traffic classification, the more conducive to network monitoring. With the widespread adoption of encryption technologies such as TLS [23], most traffic today is transmitted encrypted. Encrypted traffic is a double-edged sword, protecting privacy while providing a breeding ground for hidden malicious traffic, and putting forward higher requirements for detection systems. Fine-grained classification of encrypted traffic and detection of malicious traffic from it is a task worth investigating.

The contributions of this paper are as follows.

- By transforming traffic into images, not only making the system possible to visualise the difference between different flows, but also solving the traffic classification problem with advanced image classification techniques.

- By introducing a generative adversarial network to increase the diversity of data, and solve the problem of a small number of labeled samples in a dataset.

- By combining CNN and LadderNet as a detector, and using the multi-instance learning idea for training, the fine-grained classification of encrypted malicious traffic is realized, and the problem of low classification accuracy is solved.

The rest of the paper is organized as follows. Section 2 introduces related work and techniques. Section 3 describes the system architecture and the implementation of the algorithm. Section 4 presents some experiments of comparison between the proposed method and other encrypted malicious traffic detection algorithms. Section 5 provides conclusions and future work.

2. Related Work

2.1 Encrypted malicious traffic detection

Rule-based traffic classification methods include port-based and DPI (Deep Packet Inspection)-based methods, mainly by extracting key information such as ports or payloads in network packets according to the corresponding mapping rules for matching verification, which are more mature in application. But with the emergence of technologies such as traffic encryption and increased complexity of traffic data, rule-based detection methods gradually fail to meet today's network traffic classification and identification needs.

Traffic classification methods based on classical machine learning algorithms have been subject to much research over the past decade or so, however, there are many difficulties and drawbacks in practical applications, such as very complex feature engineering driven by domain experts, poor performance in large data scenarios, and poor real-time and adaptive performance. Unlike most traditional machine learning algorithms, deep learning networks are able to perform automatic feature extraction without human intervention, which makes them a more desirable method to classify encrypted traffic.

One of the most important advantages of deep learning is the replacement of hand-designed features with effective algorithms for feature learning and hierarchical feature extraction. The process of classifying traffic based on deep learning is divided into three steps in [2]. First, model inputs are defined and designed based on the raw packet or traffic statistics features. Then, the appropriate model and algorithm are selected based on the characteristics of the model and the purpose of the classifier. Finally, the deep learning network is trained to automatically extract the traffic features and associate the inputs with the corresponding category labels.

Lopez-Martin et al. were the first to apply recurrent neural networks (RNN) and convolutional neural networks (CNN) to the network traffic application layer identification problem [3]. Wang et al [4] proposed an end-to-end encrypted traffic classification method based on one-dimensional convolutional neural networks (1D-CNN). Gao et al in [5] proposed a deep belief network based model and Javaid et al. [6] proposed a malware traffic identification method using sparse autoencoders. All of these studies have contributed to the application of deep learning techniques to the design of network intrusion detection systems (NIDS).

The above studies all belong to two basic learning strategies for deep learning: one is supervised methods such as BP neural networks, decision trees, and support vector machines, where supervised learning performs well but has high requirements for labelling of samples during training; the other is unsupervised methods such as PCA and K-Means clustering, where unsupervised learning does not require data labelling but the performance of the models is relatively poor. In [7], a weakly supervised learning between the above two learning strategies is proposed, which is classified into three types according to the different cases of weakly labelled samples: incomplete supervised learning, inexact supervised learning, and inaccurately supervised learning. Under the weakly supervised learning strategy, the cost of dataset sample labelling is greatly reduced, which is more suitable for realistic scenarios with large amounts of data.

Therefore, this paper uses a weakly supervised learning strategy based on raw traffic data for the task of classifying encrypted traffic for malware.

2.2 Generative adversarial network

Generative adversarial network (GAN) [8] offers a promising approach to learning deep representations without the need for large amounts of annotated training data. Due to the success of GAN in applications such as computer vision and natural language processing, the technique has only been applied to the field of cyber security in recent years.

In traffic classification, several researchers have proposed methods to generate traffic samples using GAN and its improved algorithms to address the difficulties of sample labelling. A traffic data enhancement method, PacketCGAN, is proposed in [9], which can take advantage of CGAN to generate specified samples conditional on the type of incoming applications that are fed, overcoming the limitation of network data imbalance. The FlowGAN proposed in [12] trains the generated encrypted traffic data samples in combination with real data to improve the performance of application recognition on small samples. In [10], the authors propose ByteSGAN and embed it in an SDN (Software Defined Network) edge gateway, a technique that achieves the goal of flow classification in a fine-grained manner, further improving classification performance and network resource utilisation.

In this paper, we use the DCGAN architecture proposed by Radford et al. in [11], which combines CNN in supervised learning and GAN in unsupervised learning, replacing the generator and discriminator in the original GAN with two convolutional neural networks, and making some changes to the structure of the convolutional neural network to improve the quality of the samples and the speed of convergence. We also refer to the training idea of SGAN in [13], which is different from the binary classification model of the original GAN, but becomes multi-classification, and trains both the generator and the semi-supervised classifier to achieve a generative model with synthetic data closer to the real data, and a better semi-supervised classifier.

2.3 Multiple-instance learning

The concept of multiple-instance learning (MIL) was first introduced in [14], and it is often used in combination with support vector machines to form MI-SVM [15] or with convolutional neural networks to form the MIL-CNN [16] framework. MIL-based techniques have been widely used in a variety of scenarios, such as MIL-Boost [17] for target tracking, MCIL [18] for medical diagnosis, and MILCUT [19] for natural image segmentation.

In the area of malware detection, the approach presented in [20] describes the implementation of a traffic classifier as a superposition of two multi-example learning problems and shows how to map the multi-example learning problem to neural network architecture. The MIL-5min classifier proposed in [21] classifies URLs based on all network traffic observed within a 5-minute window, allowing for the training of coarse labels.

One of the advantages of multiple-instance learning is that it reduces the amount of manual labelling of data. Weakly labelled data is a prominent problem in medical imaging, where images described by a single label (benign/malignant) are often learned by multiple examples to find regions of interest (ROI). Inspired by this we integrate multi-exemplar learning into a convolutional neural network (CNN) that can learn an exemplar-level classifier to make predictions about malicious traffic and pinpoint the location of malicious traffic in the original traffic packet.

3. System Architecture

As shown in Figure 1, the system architecture of our work consists of three modules, including a data preprocessing module, a data annotation module, and a malicious traffic detection module. The following is a detailed introduction to each module.

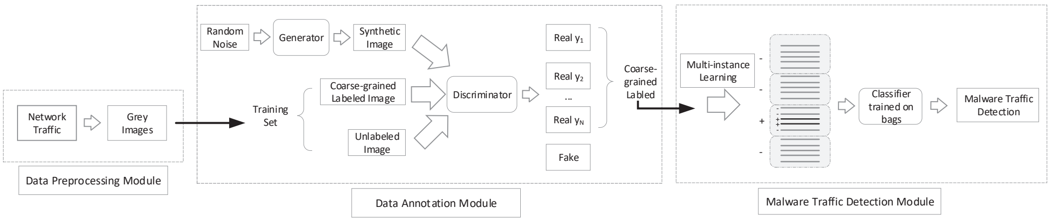

Figure 1: The system architecture of proposed method.

3.1 Data Preprocessing Module

The captured raw traffic data is usually stored in pcap file format and the file sizes are not uniform, so these raw files cannot be fed into the neural network directly. We need to preprocess the pcap files, split them into a uniform size, and convert them to several grey-scale images that could be used for traffic classification by the subsequent deep learning network model. The data preprocessing module consists of three steps: traffic split, traffic clean, and image conversion. Figure 2 shows the procedure of preprocessing.

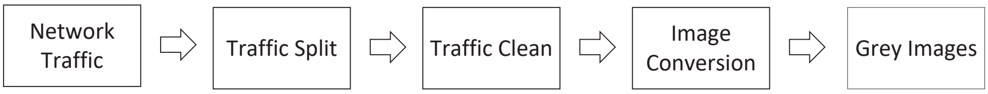

Figure 2: The steps of data preprocessing module.

Traffic Split The purpose of this step is to convert the raw flow to small, workable units at the functional level. The raw traffic data is made up of some pcap files, each of which consists of a set of packets of different sizes. The pcap file can be denoted as $P = \{p_1, p_2, ..., p_n\}$, where n is the number of packets in the file. Each packet of the pcap file is denoted as $p_i = (x_i, l_i, t_i)$, where x_i is the five-tuple given by (source IP, destination IP, source port, destination port, transport protocol) which could identify a specific packet, l_i is the byte length of the packet, and t_i is the start time of the packet transmission.

The two types of split granularity often used in traffic classification studies are flow and session. It is defined that a group of packets with the same five-tuple belong to a flow, which is denoted as $F = \{p_1 = (x_1, l_1, t_1), p_2 = (x_2, l_2, t_2), ..., p_n = (x_n, l_n, t_n)\}$, where $x_1 = x_2 = ... = x_n$, $t_1 < t_2 < ... < t_n$. A session includes all packets of the bidirectional flow. It is denoted as $S = \{p_1 = (x_1, l_1, t_1), p_2 = (x_2, l_2, t_2), ..., p_n = (x_n, l_n, t_n)\}$, where in $x_i, i = 1, 2, ..., n$, the source IP and destination IP can be interchanged. Previous studies [24] have shown that session-level recognition is effective in detecting abnormal traffic, and session can contain more information than flow. So in this step, we split the continuous raw traffic into small pcap packets based on the session.

The open-source tool SplitCap is used for traffic splitting. It is designed to split captured PCAP files into smaller files based on a criterion, such as IP address, five-tuple, or MAC address. In this system, we have used SplitCap to split raw traffic into multiple small pcap packets according to the five-tuple. This way we will get a TCP or UDP session in a separate small pcap file after splitting.

Traffic Clean This step removes some small sessions because they don't contain enough payload sizes and do little help to traffic classification. Some assisted packets should be removed as well. For example, Domain Name Service (DNS) packets that are used for hostname resolution are useless for traffic classification. So as to TCP sessions, the SYN packets, ACK packets, and FIN packets contain no payload in the transport layer. These redundant packets should be removed to reduce the subsequent workload. What's more, MAC address and IP address cannot be used as training features in traffic classification, and these information will interfere with the feature extraction process of the neural network and may cause the model overfitting, so they must be removed at random. This process is named traffic anonymisation [26].

Image Conversion In order to ensure a uniform data length, each pcap file after data cleaning is cropped to a group of 784 (28*28) bytes [27]. If the file is larger than 784 bytes, then only the first 784 bytes are retained; these bytes already contain enough payload for classification. When the file's length is insufficient, we pad with 0x00 at the end.

Each byte is an integer between 0 and 255 and is represented as a pixel, thus transforming each byte into a single-channel grey-scale image of size 28*28. Therefore, the intrinsic characteristics of the traffic can be reflected in these images. Figure 3 shows some gray-scale images of different application types after traffic conversion. It can be seen that the images of different types of application traffic have a degree of distinction.

Finally, the file containing all the pixel sequences is converted into an IDX format file for input into the deep learning model. The IDX file format is a simple format for vectors and multidimensional matrices of various numeric types, and it is also a standard input format for many neural networks such as CNN.

3.2 Data Annotation Module

For the problem that the dataset only contains a small number of labeled samples, it is necessary to expand and label the samples.

As shown in Figure 4, the data annotation module is based on a modified semi-supervised GAN, consisting of two networks, the generator G and the discriminator D. The generator G receives a random noise vector z to generate synthetic samples $x^* = G(z)$. We input the generated samples x^*, the labeled samples and their labels in real data (x, y) and the unlabeled samples x together into the discriminative network D. The discriminator is able to learn the probability of distribution of each type of data. The discriminator in the original GAN structure can only implement binary classification, determining whether the data is true or fake, which is belonging to

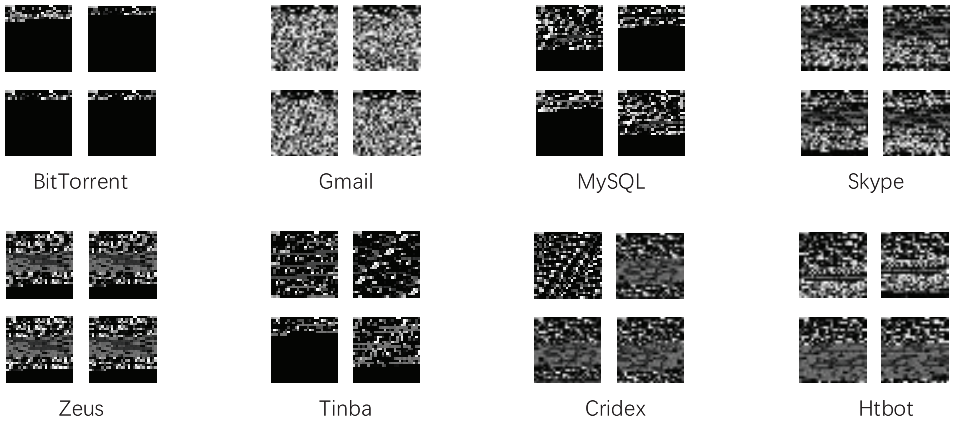

Figure 3: Visualisation of different application traffic.

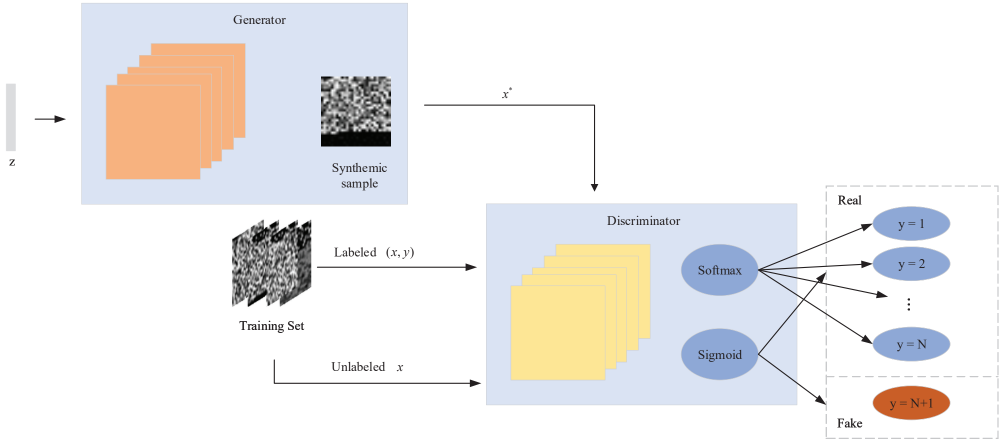

Figure 4: The network architecture of data annotation module.

unsupervised learning. The semi-supervised learning generative adversarial network (SSLGAN) then implements an $N + 1$ class classifier after the discriminator, where the first N classes are those contained in the original dataset and the $N + 1$th class are the generated samples. From this, it can not only distinguish whether the data comes from the real environment or the generator synthesis, but also complete the multi-classification task.

From the above procedure, it can be seen that the loss function of a semi-supervised GAN consists of three components. For the labelled samples in the training set, we consider the probability that the predicted label is in the corresponding class. For the unlabelled samples in the training set, the key point is the probability that the predicted label is not in the $N + 1$th class. And for the generated samples, we consider the probability that the predicted label is in the $N + 1$th class. The loss functions of three components are shown from Equation(1) to Equation(3), where p_{data} is the distribution probability of true samples, p_G is the distribution probability of generated samples and

p_{model} is the probability of the predicted classification.

$$L_{labeled} = E_{x,y \sim p_{data}(x,y)}[log p_{model}(y|x, y < N + 1)] \tag{1}$$

$$L_{unlabeled} = E_{x \sim p_{data}(x)}[log(1 - p_{model}(y = N + 1|x))] \tag{2}$$

$$L_{generated} = E_{x \sim p_G(x)}[log p_{model}(y = N + 1|x)] \tag{3}$$

The generator G and the discriminator/classifier D/C are jointly trained, and the network parameters are updated iteratively until the Nash equilibrium is reached and the network converges. The overall optimization objective function of this module is expressed as Equation(4):

$$L_{SSLGAN}(G, D) = \min_{G} \max_{D}(L_{labeled} + L_{unlabeled} + L_{generated}) \tag{4}$$

3.3 Malware Traffic Detection Module

For the problem that some of the labels in the dataset are coarse-grained, we use multiple-instance learning (MIL) in the detection module. The architecture of this module is shown in Figure 5.

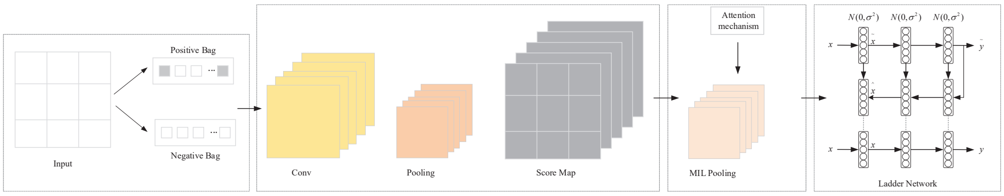

Figure 5: The network architecture of malware traffic detection module.

According to the idea of MIL, if there is at least one positive instance in a bag, which means a pcap packet contains at least one piece of malicious traffic, then the bag is labeled as positive; otherwise the pcap packet contains all normal traffic and the bag is marked as negative. The purpose of MIL is to classify the instances contained in the bag by using the labeled bags as training data. After classification of semi-supervised GAN, each small pcap packet is transformed into an image X_i which corresponds to a label Y_i. Each image $X_i = \{x_i^1, x_i^2, ..., x_i^m\}, i = 1, 2, ..., M$ is considered to be a bag in MIL.

To improve the accuracy of fine-grained classification, we combine Convolutional Neural Networks with Ladder Networks to form ConvLaddernet. We use the convolutional layer and pooling layer of CNN to receive images that have been coarsely classified by the data annotation module and output a feature map corresponding to the classification scores of different application categories to form a score map. The Ladder Network is an L-layer encoder-decoder structure that outputs two predictive labels, with the output label $\hat{y}$ of the noisy encoder used to calculate the loss function and the output label y of the noiseless encoder used for classification.

According to the structure of the multiple-instance ConvLaddernet neural network, the distribution probability of the classifier output prediction category can be calculated as Equation(5). w_i^l is the weight between the l layer and $l-1$ of the Ladder Network, and h_i^l is the hidden variable of the l layer of the noiseless encoder.

$$p(\hat{y}=y_i|x_i^j)=\frac{exp((w_i^L)^T h_i^L)}{\sum_{l=0}^{L} exp((w_i^l)^T h_i^l)} \tag{5}$$

Equation(6) to Equation(8) show the loss function of this module. The total loss function as follows also consists of two components, the supervised loss, and the unsupervised loss. λ_l represents the weight of each decoding layer's loss function. z_i^l denotes the intermediate layer output of the noiseless encoder, and $\widehat{z_i^l}$ denotes the intermediate layer output which is reconstructed by the decoder.

$$L_{supervised}=-\frac{1}{M}\sum\nolimits_{i=1}^{M}\log(p(\hat{y}=y_i|x_i^j)) \tag{6}$$

$$L_{unsupervised}=\sum\nolimits_{l=0}^{L}\frac{\lambda_l}{Mq_l}\sum\nolimits_{i=1}^{M}||z_i^l-\widehat{z_i^l}||^2 \tag{7}$$

$$L_{total}=L_{supervised}+L_{unsupervised} \tag{8}$$

4. Experiment

4.1 Datasets

Considering that there are no publicly available traffic datasets that satisfy both the encryption and malicious conditions, we conduct experiments on two datasets respectively to verify the effectiveness of the proposed method.

The dataset ISCX VPN-non VPN 2016, which contains twelve types of regular encrypted and Virtual Private Network (VPN) encrypted traffic, is used to validate the performance of encrypted traffic classification. The details of the ISCX dataset are listed in Table 1.

The dataset USTC-TFC2016, which contains twenty types of benign and malware traffic, is used to validate the performance of malicious traffic identification. The details of the dataset are listed in Table 2.

4.2 Experimental Setup

We use Python 3.6.5, TensorFlow 1.14.0 as software frameworks which run on Ubuntu 20.04 LTS OS. The Nvidia Corporation Tesla T4 GPU is used as the accelerator for computing. We use the tools in sklearn 0.22 to evaluate our model.

We use four metrics to evaluate model performance: Accuracy, Precision, Recall, and F1-score. Specific calculation methods are as Equation(9) to Equation(12). Where TP (True Positive) and FP (False Positive) are the number of instances that were correctly and incorrectly determined to be positive, and TN (True Negative) and FN (False Negative) are the number of instances that were correctly and incorrectly determined to be negative respectively.

Table 1: ISCX VPN-non VPN 2016

Dataset	Encryption Type	Traffic Type	Application
ISCX VPN-non VPN 2016	Regular encrypted traffic	Chat	Facebook, AIM, ICQ, Skype
		Email	Email, Gmail
		File Transfer	FTPS, Skype, SFTP, SCP
		P2P	Bittorrent
		Streaming	Netflix, Youtube, Vimeo, Spotify
		VoIP	Facebook, Hangouts, Voipbuster, Skype
	VPN encrypted traffic	VPN-Chat	Facebook, AIM, ICQ, Skype
		VPN-Email	Email
		VPN-File Transfer	FTPS, Skype, SFTP
		VPN-P2P	Bittorrent
		VPN-Streaming	Netflix, Youtube, Vimeo, Spotify
		VPN-VoIP	Facebook, Hangouts, Voipbuster, Skype

Table 2: USTC-TFC2016

Dataset	Traffic Type	Application
USTC-TFC2016	Benign traffic	BitTorrent, Facetime, FTP, Gmail, MySQL, Outlook, Skype, SMB, Weibo, WorldOfWarcraft
	Malware traffic	Cridex, Geodo, Htbot, Miuref, Neris, Nsis-ay, Shifu, Tinba, Virut, Zeus

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{9}$$

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

$$F1 - score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{12}$$

4.3 Experiments of comprision

We designed three comparative experiments. The four methods for comparison are Multi-Layer Perception (MLP), which is a representative of the classical machine learning method; Transfer Learning [25], which is a semi-supervised learning method; our weakly-supervised method; and CNN on behalf of supervised learning.

In experiment 1, one compares four metrics of the above methods: Accuracy, Precision, Recall, and F1-score. These four metrics are commonly used for classification tasks in machine learning. Table 3 shows the metric comparison results between the proposed model and other detection methods.

Our proposed method is optimal in three metrics Precision, Recall, and F1-score, representing good predictive and generalization performance of the model. The next best performer on these metrics is the CNN model, with the Transfer Learning and MLP being somewhat worse. As for the metric of Accuracy, CNN is slightly higher than our proposed method, since supervised learning requires more data and more accurate label information. The accuracy of the remaining two algorithms is much lower. From the metrics of dataset USTC-TFC2016, it can be seen that our method achieves an accuracy of 0.972, which is very close to the 0.979 of supervised CNN, while the quality of data labels is less demanding. Overall, the dataset USTC-TFC2016 performs better on various metrics than the dataset ISCX VPN-non VPN 2016, probably because the former classifies traffic at the application level, but the latter only classifies it at traffic type.

Table 3: Exp1. Metrics comparisons between the proposed model and other detection methods

Dataset	**Method**	**Metrics**			
		Accuracy	Precision	Recall	F1-score
ISCX VPN-non VPN 2016	MLP	0.706	0.77	0.75	0.76
	Transfer Learning	0.845	0.83	0.71	0.77
	Our Method	0.947	**0.86**	**0.81**	**0.83**
	CNN	**0.976**	0.82	0.80	0.81
USTC-TFC2016	MLP	0.725	0.87	0.88	0.87
	Transfer Learning	0.764	0.90	0.86	0.88
	Our Method	0.972	**0.95**	**0.93**	**0.94**
	CNN	**0.979**	0.92	0.89	0.91

In experiment 2, we want to study the effect of different percentages of labeled samples on the accuracy of different methods. We design to test the accuracy of the proposed model and other detection methods when the labeled samples account for 1%, 5%, 10%, 50% and 100% of the training samples respectively. Figure 6 shows line graphs of the results of experiment 2.

As shown in Figure 6, the classification accuracy of MLP is significantly lower than that of the remaining three neural network models, and the accuracy of Transfer Learning is also relatively low, which is below 0.85. As we mentioned in Experiment 1, in the accuracy experiments with 100% labeled samples, the CNN is more accurate. However, our method could also achieve relatively high accuracy, especially in USTC-TFC2016, where the accuracy of the two is very close. In the experiment with 50% labeled samples, there are highs and lows between our method and CNN. While in experiments with 1%, 5% and 10% labeled samples, our method was significantly more accurate than the other methods, and the accuracy rate can generally reach more than 0.90 on both datasets. However, in experiments where the proportion of labeled samples was less than 10%, the accuracy of the CNN begin to decrease gradually, falling below 0.90. It shows that our model outperforms CNN at low percentage of labeled samples.

To further explore the performance of training with fewer percentage of labeled samples, we designed an accuracy comparison experiment under 1% to 5% labeled samples, that is experiment 3. Figure 7 shows a visualisation of the results of the accuracy comparison between the proposed

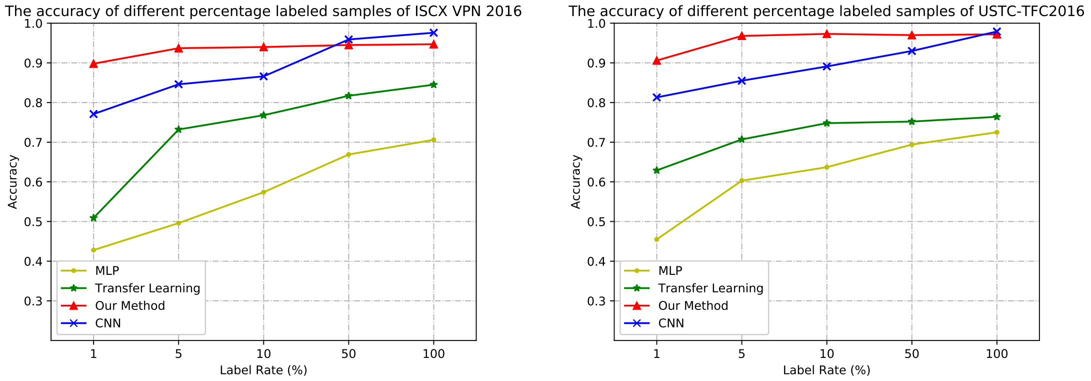

Figure 6: Accuracy comparison results of different label rates.

model and other detection methods.

It can be seen that the accuracy of our method is significantly higher than the other three methods when the label rate of samples is under 5% . Our method could achieve approximately 0.90 accuracy on both datasets in the 1% labeled sample experiment, there are 0.898 and 0.906 respectively. When the proportion of labeled samples is more than 3%, the accuracy rate on the dataset ISCX can reach more than 0.93, and that of the dataset USTC-TFC2016 even over 0.95. Whereas the other models are all below 0.90, and the accuracy decreases significantly as the proportion of labeled samples decreases. It turns out that our model has better performance on a small number of labeled samples.

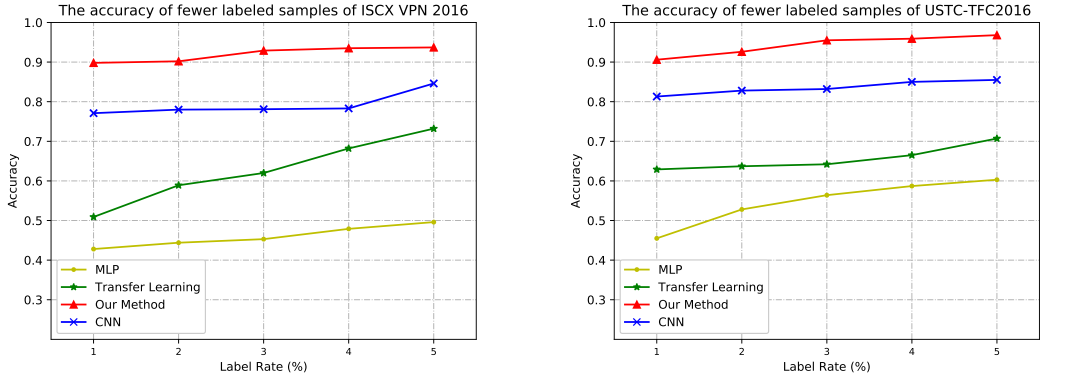

Figure 7: Accuracy comparison results of fewer label rates.

5. Conclusion

In this paper, we propose a weakly-supervised method that could extract features, select features of encrypted network traffic, and classify encrypted network traffic automatically. Compared with traditional malicious traffic classification countermeasures, the proposed detection method could achieve the fine-grained classification of encrypted malicious traffic with a small number of coarse-grained labeled samples and a large number of unlabeled samples. And it has higher accuracy, especially with a fewer labeled sample proportion.

We expect to deploy this approach on IHEPSOC later to improve its detection performance. However, there are still so many problems in real traffic, such as traffic imbalance, real-time detection, and cascading encrypted traffic, which may be the direction we want to explore in the future.

Acknowledgments

This work is supported in part by the Xiejialin Project of Institute of High Energy Physics under Grant no. E25467U2 and the specialized project for cybersecurity and informatization in the 14th Five-Year Plan of CAS under grant no. WX145XQ12 and National Natural Science Foundation of China under grant no. 61901447

References

[1] Wang J, Yan T, An D, et al. A comprehensive security operation center based on big data analytics and threat intelligence[C]//International Symposium on Grids & Clouds 2021. 22-26 March 2021. Academia Sinica Computing Centre (ASGC). 2021: 28.

[2] Wang P, Chen X, Ye F, et al. A survey of techniques for mobile service encrypted traffic classification using deep learning[J]. IEEE Access, 2019, 7: 54024-54033.

[3] Lopez-Martin M, Carro B, Sanchez-Esguevillas A, et al. Network traffic classifier with convolutional and recurrent neural networks for Internet of Things[J]. IEEE access, 2017, 5: 18042-18050.

[4] 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.

[5] Gao N, Gao L, Gao Q, et al. An intrusion detection model based on deep belief networks[C]//2014 Second International Conference on Advanced Cloud and Big Data. IEEE, 2014: 247-252.

[6] Javaid A, Niyaz Q, Sun W, et al. A deep learning approach for network intrusion detection system[J]. Eai Endorsed Transactions on Security and Safety, 2016, 3(9): e2.

[7] Zhou Z H. A brief introduction to weakly supervised learning[J]. National science review, 2018, 5(1): 44-53.

[8] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[J]. Advances in neural information processing systems, 2014, 27.

[9] Wang P, Li S, Ye F, et al. PacketCGAN: Exploratory study of class imbalance for encrypted traffic classification using CGAN[C]//ICC 2020-2020 IEEE International Conference on Communications (ICC). IEEE, 2020: 1-7.

[10] Wang P, Wang Z, Ye F, et al. ByteSGAN: A semi-supervised Generative Adversarial Network for encrypted traffic classification in SDN Edge Gateway[J]. Computer Networks, 2021, 200: 108535.

[11] Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks[J]. arXiv preprint arXiv:1511.06434, 2015.

[12] Wang Z X, Wang P, Zhou X, et al. FLOWGAN: Unbalanced network encrypted traffic identification method based on GAN[C]//2019 IEEE Intl Conf on Parallel Distributed Processing with Applications, Big Data Cloud Computing, Sustainable Computing & Communications, Social Computing Networking (ISPA/BDCloud/SocialCom/SustainCom). IEEE, 2019: 975-983.

[13] Odena A. Semi-supervised learning with generative adversarial networks[J]. arXiv preprint arXiv:1606.01583, 2016.

[14] Dietterich T G, Lathrop R H, Lozano-Pérez T. Solving the multiple instance problem with axis-parallel rectangles[J]. Artificial intelligence, 1997, 89(1-2): 31-71.

[15] Andrews S, Tsochantaridis I, Hofmann T. Support vector machines for multiple-instance learning[J]. Advances in neural information processing systems, 2002, 15.

[16] Sun M, Han T X, Liu M C, et al. Multiple instance learning convolutional neural networks for object recognition[C]//2016 23rd International Conference on Pattern Recognition (ICPR). IEEE, 2016: 3270-3275.

[17] Babenko B, Dollár P, Tu Z, et al. Simultaneous learning and alignment: Multi-instance and multi-pose learning[C]//Workshop on Faces in’Real-Life’Images: Detection, Alignment, and Recognition. 2008.

[18] Xu Y, Zhu J Y, Eric I, et al. Weakly supervised histopathology cancer image segmentation and classification[J]. Medical image analysis, 2014, 18(3): 591-604.

[19] Wu J, Zhao Y, Zhu J Y, et al. Milcut: A sweeping line multiple instance learning paradigm for interactive image segmentation[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014: 256-263.

[20] Pevny T, Somol P. Discriminative models for multi-instance problems with tree structure[C]//Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security. 2016: 83-91.

[21] Pevny T, Dedic M. Nested multiple instance learning in modelling of HTTP network traffic[J]. arXiv preprint arXiv:2002.04059, 2020.

[22] Rezaei S, Liu X. Deep learning for encrypted traffic classification: An overview[J]. IEEE communications magazine, 2019, 57(5): 76-81.

[23] Velan P, Čermák M, Čeleda P, et al. A survey of methods for encrypted traffic classification and analysis[J]. International Journal of Network Management, 2015, 25(5): 355-374.

[24] Cui S, Jiang B, Cai Z, et al. A session-packets-based encrypted traffic classification using capsule neural networks[C]//2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS). IEEE, 2019: 429-436.

[25] Rezaei S, Liu X. How to achieve high classification accuracy with just a few labels: A semi-supervised approach using sampled packets[J]. arXiv preprint arXiv:1812.09761, 2018.

[26] 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, 2006, pp. 2302-2309, doi: 10.1109/ICC.2006.255113.

[27] Wei Wang, Ming Zhu, Xuewen Zeng, Xiaozhou Ye and Yiqiang Sheng, "Malware traffic classification using convolutional neural network for representation learning," 2017 International Conference on Information Networking (ICOIN), 2017, pp. 712-717, doi: 10.1109/ICOIN.2017.7899588.