PROCEEDINGS OF SCIENCE

Malicious Traffic Detection with Class Imbalanced Data Based on Coarse-grained Labels

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

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

[b] *School of Cyber Science and Engineering, Zhengzhou University, Zhengzhou 450001, China*

E-mail: wangjr@ihep.ac.cn

In order to resist complex cyber-attacks, a Security Operations Center (SOC) named IHEPSOC has been developed and deployed in the Institute of High Energy Physics (IHEP) of the Chinese Academy of Sciences, which contributed to the reliability and security of the network for IHEP. It has become a major task to integrate state-of-the-art cyber-attack detection methods for IHEPSOC to improve the ability of threat detection. Malicious traffic detection based on machine learning is an emerging security paradigm, which can effectively detect both known and unknown cyber-attacks. However, the existing studies usually adopt traditional supervised learning, which often encounter issues when applied to real-world production environment due to its implicit assumptions on the operating dependence. For example, most studies are based on datasets that already have accurate data labels, but labeling these datasets accurately requires significant manual effort. In addition, in the real-world service, the volume of benign traffic data is larger than that of the malicious traffic data, and the imbalance between benign and malicious categories also makes many machine learning detection models difficult to apply to a production environment. Based on these, we propose a detection method for class imbalanced malicious traffic based on coarse-grained data labels, which achieves comparable performance compare to other supervised learning methods. We conducted three experiments, using the Android Malware 2017 dataset, and verified the practicability and effectiveness of the proposed method.

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

*Speaker

PoS(ISGC2022)030

1. Introduction

Malicious traffic includes network attacks, phishing, malicious crawlers, and so on, which usually intrudes and grabs other parties' services or data through unauthorized means. Preventing the network from being attacked by malicious traffic is important to ensure the normal operation of people's work and life. IHEPSOC has conducted intensive research and exploration in the field of network security. It has been found that in this field, machine learning, especially deep learning methods, have achieved significant progresses in recent years [1]. Machine learning is a technology that can map data from the original feature space to the target label space, which usually consists of a model with initial parameters and a data set. The mapping relationship is approximated by a specific optimization algorithm to achieve the purpose of predicting the desired result. Compared with the rule-based malicious traffic detection method, machine learning method can better extract the encrypted information, timing relationship and other complex features in network traffic [2]. However, most research related to deep learning implicitly assumes that there is a large amount of data with accurate labels, it's not feasible to obtain such data in the practical production environment for many organizations. Obtaining a large amount of network traffic data with accurate labels takes significant human resources, which makes many otherwise appropriate theoretical models difficult to apply in practice. The weakly supervised learning method can attain the expected machine learning model on the premise of reducing the requirements for label quality or quantity [3], which makes such models better applicable to practical production. In addition, the network traffic data often has a huge imbalance problem, that is, the amount of normal network traffic is far exceeds that of malicious network traffic, which will also hinder many models from achieving their optimal discriminating power. Especially when the data imbalance is combined with weak supervised learning, both problems will become more difficult to solve.

In this paper, we aim to address both problems identified above, the malicious traffic detection is modelled as a multi-instance learning (MIL) problem, and the corresponding solution to the data imbalance problem is proposed. The proposed method only needs network traffic data with time-slice-level coarse-grained labels for training to attain a high practical and accurate deep learning model. Specifically, as shown in Figure 1, the proposed method takes the data with coarse-grained label of time slice dimension as the input, uses multiple clustering algorithms to pre-estimate the data in the time slice containing malicious traffic in multiple subspaces, and then transmits it to the backbone network for training and prediction. Finally, the pre-estimated score is updated according to the prediction results. The specific process is introduced in Part 3. The proposed method has been deployed on IHEPSOC, which effectively detects the malicious traffic in IHEP and contributes guarantee for the network security of IHEP.

The contributions of this paper are as follows:

- We proposed a weak supervised learning method for network malicious traffic detection, which can obtain a deep learning model with high practicability and high accuracy even when only the data with coarse-grained labels are used. Because the backbone neural network of the proposed method can be replaced, it means that many existing supervised learning achievements can be combined with the proposed method, in order to meet the practical production needs.

- A solution to the imbalance problem under weakly supervised learning is proposed. Several clustering models are used to pre-estimate the malicious score of traffic data, and the malicious score is dynamically adjusted by the neural network. In this way, the model can still pay more attention to the relative small fraction of malicious malicious traffic data.

- Based on the public dataset Android malware 2017 [21], the coarse granularity of data labels is used for the proposed method. Through the imbalance experiment, contrast experiment and ablation experiment, the practicability, accuracy and effectiveness of the proposed method are verified.

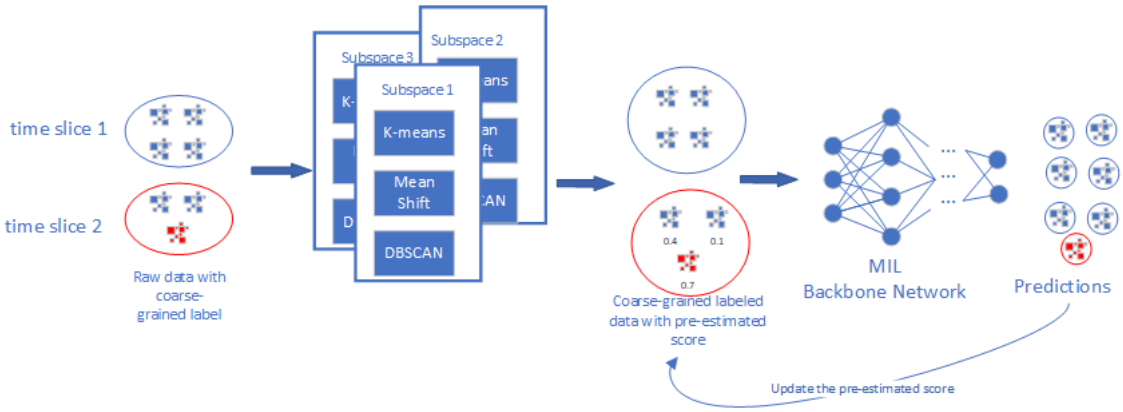

Figure 1: Pipeline of the proposed method. Firstly, the coarse-grained labeled data are fed into several clustering models in different subspace to get the pre-estimated score. Then, training the backbone network with these coarse-grained labeled data. Finally, update the pre-estimated score according to the predictions.

2. Related works

Network traffic detection methods can generally be divided into rule-based and machine learning-based. In the rule-based method, the method of comparing traffic is to use Deep Packet Inspection (DPI) for traffic classification. For example, Wang et al. [4] proposed a scalable regular expression matching method, which extracts a much shorter stride-length stream from the original byte stream convert for regex matching and greatly improves the speed of DPI detection technology. The open-source tool nDPI [5] is also very effective in many scenarios. It is composed of a core library and plugin dissectors. It has rich protocol support and better scalability. More DPI-related work can be found in related reviews [6][7]. Due to the diversification of attack methods and the complexity of encrypted traffic, it is difficult for such rule-based methods to capture the deep characteristics of network traffic, which limits its detection capability. The detection technology based on a machine learning method has a better extraction ability for the in-depth features of network traffic. Wang et al. [8] first designed a Convolutional Neural Networks (CNN) structure for malicious network traffic detection by using the way of characterization learning. It did not need to design the features manually, but directly took the raw traffic as the input, and has achieved good detection performance. Gao et al. [9] designed a two-level anomaly detection system, which uses an Apriori association algorithm to mine the association rules between the discretized features and

the "normal" label to filter out the normal traffic misclassified by neural network and achieved a low false positive rate on NSL-KDD [22] dataset. Fu et al. [10] also optimized the deep learning model. By extracting frequency domain features for training the deep learning model, the detection accuracy was improved, the scale of features was limited, and faster detection speed was realized. Many other studies [11] [12] [13] have also carried out performance optimization based on various deep learning models, but most of these works are based on supervised learning, which requires a large amount of data with accurate labels, which needs lots of human effort in a practical production environment.

By weakening the demand for data labels, weakly supervised learning methods can enable many models to be applied more easily. These models can usually be divided into three types: incomplete supervision, inexact supervision and inaccurate supervision [3]. Among them, the inexact supervision type of the weak supervised learning method only needs coarse-grained data labels, which is very easy to implement in the process of capturing network traffic, and the multi-instance learning (MIL) [14] method can be used to solve this kind of weak supervised learning problem. MIL has achieved good results in many scenes. For example, in the field of computer vision [15] [16], researchers regard video surveillance as a whole, which only needs video-level coarse-grained labels, and the model can learn the time-slice-level features of abnormal events. Similarly, Cances et al. [17] applied MIL to sound event detection, which was also only given coarse-grained labels of sound events without time boundary, and achieved good performance in combination with Convolutional Recurrent Neural Network (CRNN). All this research greatly reduces the cost of manually labeling data, making many models more practical.

3. Proposed Methods

In this section, we will introduce the proposed weakly supervised malicious traffic detection method in detail. There four parts in this section: data labeling, positive score pre-estimate, loss function, and positive score adjustment strategy. These methods are organized according to the pipeline of Figure 1 to realize the weakly supervised malicious traffic detection with coarse-grained labeled data.

3.1 Data Labeling

In the practical production environment, data with fine-grained labels is often difficult to obtain, since it requires a lot of manual effort. In this paper, traffic detection is modeled as a multi-instance learning problem, and the requirements for data labeling are relaxed by considering only the time-slice-level coarse-grained label. Specifically, as shown in Figure 2, experts only need to determine whether there is malicious traffic in a time slice when capturing the generated network traffic, and do not need to pay attention to where a specific malicious data item is within the larger captured time slice. When it is determined that at least one piece of data in a time slice belongs to a malicious category label, all the data in this time slice will be labeled with this category of malicious label, although most of the data is benign. If there is no malicious traffic in a time slice, the time slice is considered to be normal, and all data in this time slice will be labeled as benign.

This process is expressed in MIL as follows. A collection of all data in a time slice is called a bag, and each specific piece of data in the collection is called an instance, malicious traffic data

represents a positive instance, and normal traffic data represents a negative instance. When there is at least one positive instance in a bag, we call it a positive bag, otherwise call it a negative bag, and all instances in the bag inherit the label of the bag. This labeling method is coarse-grained in the positive bag, because only a few instances in a positive bag are true positive instances, while most instances are false positive instances. And this labeling method is fine-grained in the negative bag, because there is no positive instance in the negative bag, which means that all instances are true negative instances. This data labeling process can provide significant labor cost savings, making it affordable for many organizations.

The formal definition is as follows:

- Positive instance refers to malicious traffic, and it is denoted as i_{po}.
- Negative instance refers to benign traffic, and it is denoted as i_{ne}.
- Positive bag refers to a collection of instances in a time slice with at least one positive instance, and it is denoted as $B_{po} = \{i_{ne}^1, ..., i_{ne}^m, i_{po}^1, ..., i_{po}^l\}$, where $m >= 0$ is the number of negative instances in B_{po}, $l >= 1$ is the number of positive instances in B_{po}.
- Negative bag refers to a set of instances in a time slice without any positive instance, and it is denoted as $B_{ne} = \{i_{ne}^1, ..., i_{ne}^k\}$, where k is the number of negative instances in B_{ne}.
- And the positive label is denoted as y_{po} (this value varies by specific malicious category), the negative label is denoted as y_{ne}. All instances have the same label as their bag. Since the label of the positive bag is coarse-grained, if a positive instance i_{po} is labeled as y_{po}, we call it a true positive instance, if a negative instance i_{ne} is labeled as y_{po}, we call it a false positive instance. The target for training is to find the true positive instance in positive bag and distinguish the false positive instance.

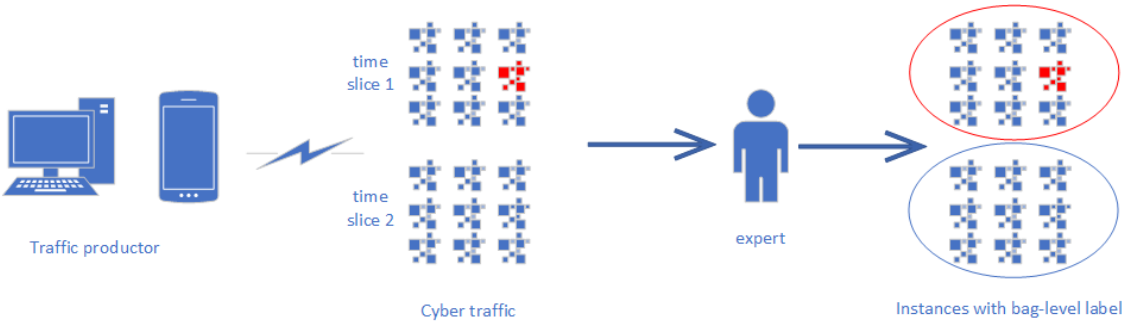

Figure 2: The process of data labeling. Experts regard all the data in a time slice as a bag, and give only bag-level label.

3.2 Positive Score Pre-Estimate

As mentioned above, in actual scenarios, the proportion of malicious traffic is much smaller than that of normal traffic, and this imbalance will have a great negative impact on the training process. In the weakly supervised learning method used in this paper, malicious traffic and normal

traffic together form a positive bag, which further exacerbates this imbalance problem. Inspired by Carbonneau et al. [18], we use multiple clustering algorithms on different subspaces to pre-estimate the positive score of instances in the positive bag.

Specifically, as shown in Figure 3, the instances in various types of positive bags will be projected in the subspace together with the instances of negative bags, and they will be clustered using a clustering algorithm. When a cluster can represent a certain malicious feature, the cluster tends to contain a relatively high proportion of true positive instances of this malicious category. Through this method, we can perform a pre-estimate operation on the instance in the positive bag. In each cluster, using the proportion of instances from a kind of positive bag in the cluster as the pre-estimated positive score for the corresponding category, instances from this kind of positive bag will inherit the cluster's pre-estimated score for this category. Multiple clustering algorithms will be used for pre-estimating in a subspace, and this process will be repeated in different subspaces. The instances from positive bag will get multiple pre-estimated positive score, and the mean of these pre-estimated positive score is final pre-estimated score for each instance. This score is only used as the initial positive score of the data, and will be dynamically adjusted during the training process as described in 3.4, so it is not sensitive to parameters such as the number of subspaces.

The formal definition is as follows:

- Collection of subspaces is denoted as $O = \{S_1, ..., S_n\}$, where S_i is a subspace, which can be obtained by techniques like dimensionality reduction, and n is the number of sub-spaces.

- Collection of clusters in a clustering process is denoted as $C = \{c_1, ..., c_n\}$, where c_i is a cluster, and n is the number of clusters.

- Collection of the pre-estimated positive score for malicious categories in a cluster is denoted as $u = \{s_1, \ldots, s_m\}$, where s_i is the pre-estimated positive score for the $i-th$ malicious category in the cluter, m is the number of malicious categories.

- Collection of the pre-estimated positive score for clusters in a clustering process is denoted as $U = \{u_1, ..., u_n\}$, where $u_i = \{s_1^i, \ldots, s_m^i\}$ is the pre-estimated positive score for malicious categories in the $i-th$ cluster, n is the number of clusters, m is the number of malicious categories.

- Collection of the final pre-estimated positive score for instances from positive bag is denoted as $P = \{E_1, ..., E_n\}$, where E_i is the mean pre-estimated score for an instance under multiple clustering, n is the number of instances from positive bag.

- Collection of clustering algorithms is denoted as $A = \{a_1, ..., a_n\}$, where a_i is a specific clustering algorithm (e.g. K-Means), n is the number of algorithms used.

The above process is expressed as Algorithm 1.

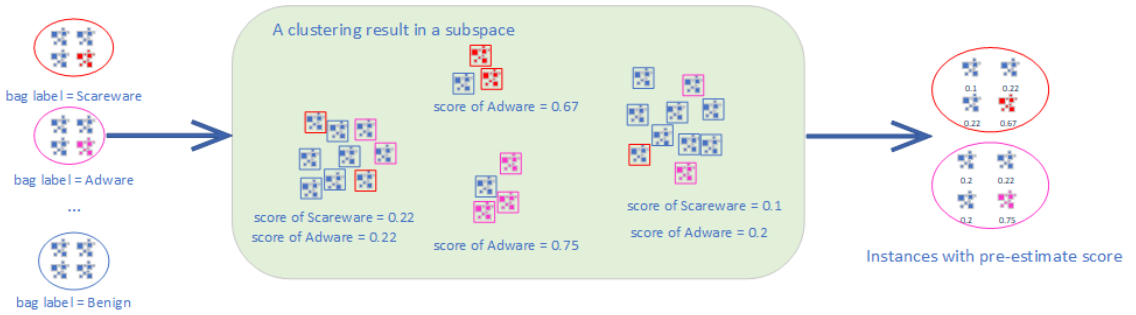

Figure 3: The process of pre-estimate the positive score in a subspace with a clustering algorithm. Every cluster calculate the positive score for each malicious category by their proportion. Instances inherit the score of their clusters.

Algorithm 1: Positive Score Pre-Estimate

Input: O,B_{po},B_{ne}
Output: P

```
foreach S_i in O do
    Project B_po,B_ne into B'_po,B'_ne on S_i
    foreach a_i in A do
        Cluster B'_po,B'_ne to form C by a_i
        Calculate U by the proportion of every malicious category C
        Instances from positive bag inherit the corresponding positive score from their
          cluster
    end
end
Calculate P for all instances from positive bag by their average positive score
```

3.3 Loss Function

Since the data only uses coarse-grained data labels, the commonly used loss function cannot evaluate the error of the instances in the positive bag during the training process. As shown in Figure 4, we determined that the instance in the negative bag will get a smaller malicious prediction value, the true positive instance in the positive bag will get a larger malicious prediction value, and the false positive instance will get a smaller malicious prediction value. In the negative bag, since the bag-level label is also fine-grained instance-level label, the cross-entropy loss function can be directly used for optimization. And in the positive bag, the instances only have coarse-grained labels, and the true positive instance cannot be found directly. But the prediction for true positive instance is more likely closer to the corresponding positive label than the false positive instance. So, we can regard the instance with prediction closest to the positive label as the true positive instance and calculate the cross-entropy loss with positive label, while other instances calculate cross-entropy loss with benign label. The cross-entropy loss function is denoted as $\psi(\cdot)$, and the prediction for an instance is denoted as p_i, with λ is a coefficient can be set, we define the following

loss function:

$$Loss = \begin{cases} \overbrace{\psi_{min}(y_{po}, p_k)}^{V_1} + \overbrace{\lambda \frac{1}{n-1} \sum_{j=1, j\neq k}^{n} \psi(y_{ne}, p_j)}^{V_2} & Positive\ bag \\ \frac{1}{n} \sum_{i=1}^{n} \psi(y_{ne}, p_j) & Negative\ bag \end{cases} \tag{1}$$
$$\psi_{min}(y_{po}, p_k) = min_{1 \leq i \leq n}(\psi(y_{po}, p_i))$$

the part V_1 in positive bag in the equation 1 describes the target of optimizing for the instance whose predicted value is closest to the positive label, which means that the $k-th$ instance has the minimum cross entropy loss with the positive label. The part V_2 in the equation 1 in positive bag describes the target of optimizing the instances except the $k-th$ one. And the instances from negative bag have the fine-grained labels, so the cross entropy loss is applied directly.

However, due to the small proportion of true positive instances in the positive bag, the model may not be able to predict the true positive instance with a prediction closest to the positive label in the bag, resulting in the model optimizing in a wrong direction. We use the pre-estimated score of instances in the positive bag as the weight. This weight will be used as an auxiliary coefficient in part V_1 in the equation 1 to determine the instance that is closest to the positive label, as follows:

$$\psi_{min}(y_{po}, p_k) = min_{1 \leq i \leq n}(\psi(y_{po}, p_k) \times (1 - E_i)) \tag{2}$$

In the equation 2, the larger the estimated positive score E_i of an instance is, the easier it is to be judged as the instance whose prediction is closest to the positive label, which means that the model can more easily pay attention to the instance with high probability of being positive.The auxiliary coefficient $(1 - E_i)$ should be eliminated in the part V_1 in the equation 1 for optimizing, we have the final loss function as follows:

$$Loss = \begin{cases} \frac{\psi_{min}(y_{po}, p_k)}{1 - E_k} + \lambda \frac{1}{n-1} \sum_{j=1, j\neq k}^{n} \psi(y_{ne}, p_j) & Positive\ bag \\ \frac{1}{n} \sum_{i=1}^{n} \psi(y_{ne}, p_j) & Negative\ bag \end{cases} \tag{3}$$
$$\psi_{min}(y_{po}, p_k) = min_{1 \leq i \leq n}(\psi(y_{po}, p_i) \times (1 - E_i))$$

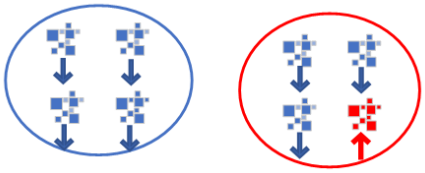

Figure 4: The target of training.

3.4 Positive Score Adjustment

If the pre-estimated score of the positive bag instance obtained in $Algorithm1$ is fixed in the loss function, the clustering algorithm used in the pre-estimation will occupy a relatively dominant

position in the training process. This will hinder the optimization of the neural network, because there too much prior knowledge given by the clustering model in training.

In order to reduce this influence, so that the neural network model can be in a dominant position and mine deeper feature information, the pre-estimated positive score of the instances should be dynamically adjusted. As shown in Figure 5, when an instance is predicted to be positive, its estimated score in the next training should be increased, otherwise, its pre-estimated score in the next training should be decreased. We control this update using sin and arcsin, which limit the score from 0 to 1 and are gentle in increasing or decreasing the score. The specific adjustment formula is as follows:

$$E_i = \begin{cases} \sin(\frac{\pi}{2} E_i) & i = k \\ \frac{2}{\pi} \arcsin(E_i) & i \neq k \end{cases} \tag{4}$$

where k corresponds to the instance that satisfy equation 2.

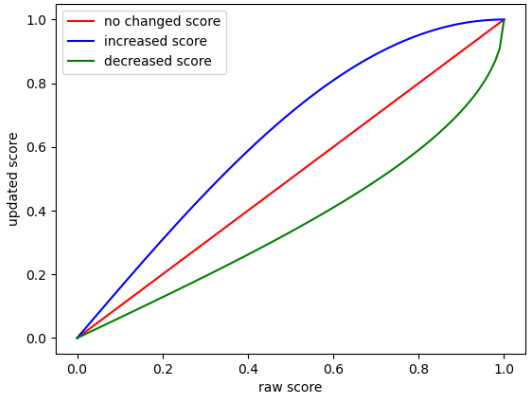

Figure 5: The strategy for positive score adjustment. When an instance is predicted to be positive, increase its positive score in the next step by *sin* (the blue line), otherwise, decrease its positive score in the next step by $arcsin$ (the green line).

4. Experiments

4.1 Setup

The Android Malware 2017 dataset is used for the experimentals, which contains five types of network traffic data: Adware, Ransomeware, Scareware, SMSmalware and Benign, with a total of about 4.67 million data items. Each data item of Android Malware 2017 has accurate labels, which can be used to train the supervised learning model as a comparative experiment of the weakly supervised learning model proposed in this paper.

To obtain coarse-grained labeled data from this dataset, this paper randomly inserts Benign type data into other malicious types of data, and then generates positive bags with a certain imbalance rate. Within Benign type, several data are randomly selected to produce negative bags. The Android Malware 2017 data with coarse-grained labels generated according to this process will be used as the training set of the method proposed in this paper.

In this paper, the CNN structure proposed by Wang et al. [8] is used as the backbone neural network. The experiment is carried out on a server equipped with NVIDIA Corporation Tesla T4 GPU, the operating system is Ubuntu 20.04 LTS, and the development environment is Python 2.7 and Tensorflow 1.12. Area under the ROC curve (AUC) is used as the evaluation indicator. AUC comprehensively considers the performance of the model on True Positive Rate (TPR) and False Positive Rate (FPR). The larger its value, the better the performance of the model.

4.2 Imbalance Rate Experiment

As mentioned above, the imbalance of data will have a great impact on the performance of the model. This paper weights the instances in positive bag by pre-estimating the positive score, so that the model pays more attention to the instances with large positive score. In order to verify the influence of the imbalance rate in the positive bag on the performance of the model, different imbalance rates are used to generate coarse-grained label data, which uses the proposed method for training. As shown in Figure 6, when the imbalance rate of the model is between 50:1 and 20:1, the performance of the model will rise obviously with the decrease of the imbalance rate, and this upward trend is close to convergence when the imbalance rate is lower than 20:1. Actually, due to the existence of also negative bags, the overall data imbalance rate is higher than that of just the positive bags. It can be seen that the model has achieved good results when the imbalance rate is 20:1, and it is plausible that experts are able to label the time slices of normal traffic and malicious traffic of 20:1 when conducting malware simulation experiments, which means that the method proposed in this paper has good practicability.

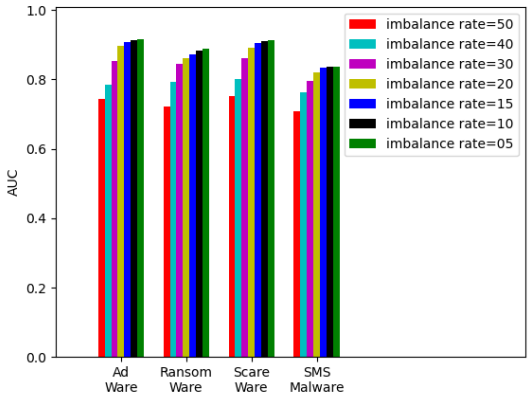

Figure 6: AUC of the imbalance rate experiment.

4.3 Contrast Experiment

In order to study the performance comparison between the proposed method and other supervised learning methods, the contrast experiment is carried out. Three supervised learning methods are used to compare with the methods proposed in this paper. They are CNN proposed by Wang et al. [8], Support Vector Machine (SVM), and Hidden Markov Model (HMM). These methods use data with accurate data labels, while our method, labelled "Proposed" in Table 1, only uses coarse-grained labels, while maintaining the imbalance rate of positive bags at 15:1. As shown in

Figure 7 and Table 1, the proposed method achieves comparable performance in the other three malicious traffic categories except Ransomware. Compared with the traditional supervised learning method and the CNN method with the same network structure that use fine-grained labels, it can be seen that although only coarse-grained labeled data are used, the proposed method can still achieve comparable performance. Using the proposed method in combination with an advanced backbone neural network structure in a real-world production environment can achieve performance close to or even better than traditional machine learning methods, and significantly save the cost of manual labeling.

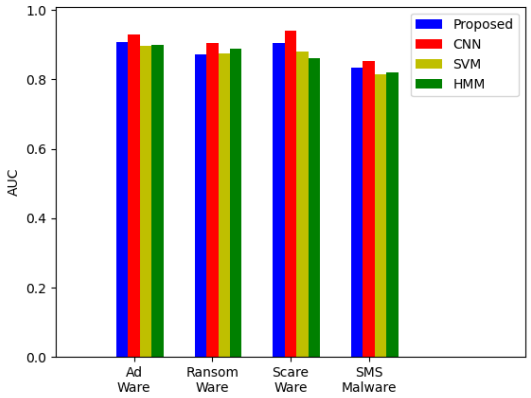

Figure 7: AUC of the contrast experiment.

Category	Proposed	CNN	SVM	HMM
Ad ware	88.72%	91.04%	87.64%	88.04%
Ransom ware	85.34%	88.37%	85.47%	86.96%
Scare ware	88.38%	92.12%	86.01%	84.02%
SMS malware	81.35%	83.16%	79.57%	80.13%

Table 1: AUC of the contrast experiment. "Proposed" is the proposed method in this work with only coarse-grained labels for training, while others are the contrast supervised methods with fine-grained labels for training.

4.4 Ablation Experiment

In order to verify the effectiveness of each module of the proposed method, an ablation experiment is conducted. Specifically, in group "MIL" only the MIL backbone neural network and unweighted loss function are used, then the clustering module is added to pre-estimate the positive score and the weighted loss function is used in group "Pre-Estimated", and finally the adjustment strategy for pre-estimated positive score is added in "Proposed". As shown in Figure 8 and Table 2, after adding the pre-estimate module to the basic MIL backbone neural network, the performance of the model has been significantly improved. This is mainly because after the clustering module

pre-estimates the instance of positive bag, the backbone network can attain a priori knowledge and pay more attention to the instance with a high probability of being a true positive. After the final addition of the estimated positive score adjustment strategy, the backbone network can not only attain the prior knowledge of the clustering model in the early stage of training, but also gradually attain the dominant position in the training process, so it further improves the performance of the model. The effectiveness of each module proposed in this paper is demonstrated.

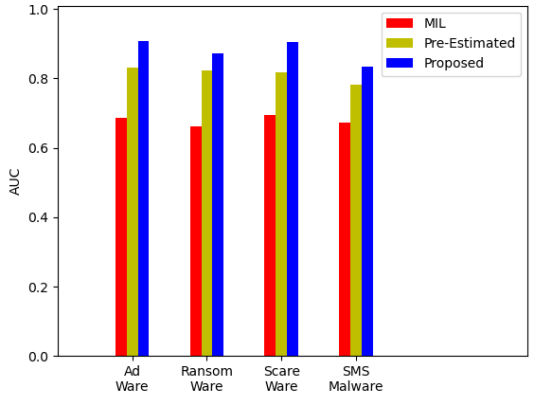

Figure 8: AUC of the ablation experiment.

Category	MIL	Pre-Estimated	Proposed
Ad ware	66.71%	81.14%	88.72%
Ransom ware	64.34%	80.42%	85.34%
Scare ware	67.43%	79.78%	88.38%
SMS malware	65.32%	76.16%	81.35%

Table 2: AUC of the ablation experiment. "MIL" is the backbone neural network without any class imbalance treatment. "Pre-Estimated" uses the pre-estimated score based on "MIL". "Proposed" uses all of the proposed class imbalance treatments based on "MIL". All three methods use coarse-grained labels for training.

5. Conclusion

Considering the dual challenge of high data labeling cost and data imbalance in malicious traffic detection, this paper proposes a weakly supervised learning model, that models malicious traffic detection as a multi-instance learning problem, and uses multiple clustering algorithms to pre-estimate the data in multiple subspaces and thereby assists the backbone network training. Finally, the backbone network attains the dominant position of prediction through the pre-estimated score adjustment strategy. The practicability, accuracy and effectiveness of the proposed model are verified through three experiments. Organizations only need to give a data label to a time slice when capturing network traffic data to train the malicious traffic detection model required

by the proposed method, and obtain a detection model with the performance equivalent to that of traditional supervised learning.

In this paper, CNN network is used as the backbone network. However, the proposed method is generic and independent of the specific backbone neural network, and can use many other neural network structures, such as Transformer [19] and its variants, as the backbone network to obtain better malicious traffic feature extraction ability. In addition, because the real network traffic data are not independent and identically distributed [20], there is often some correlation between the data under the same or similar time slices. Being able to correctly model this correlation can further improve the performance of weakly supervised learning. The further exploration in this regard will be also made.

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] Berman D S, Buczak A L, Chavis J S, et al. A survey of deep learning methods for cyber security[J]. Information, 2019, 10(4): 122.

[2] Wang Z, Fok K W, Thing V L L. Machine learning for encrypted malicious traffic detection: Approaches, datasets and comparative study[J]. Computers & Security, 2022, 113: 102542.

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

[4] Wang, X., Jiang, J., Tang, Y., Liu, B., & Wang, X. (2011, June). StriD^{2}FA: Scalable Regular Expression Matching for Deep Packet Inspection. In 2011 IEEE International Conference on Communications (ICC) (pp. 1-5). IEEE.

[5] Deri, L., Martinelli, M., Bujlow, T., & Cardigliano, A. (2014, August). nDPI: Open-source high-speed deep packet inspection. In 2014 International Wireless Communications and Mobile Computing Conference (IWCMC) (pp. 617-622). IEEE.

[6] Finsterbusch M, Richter C, Rocha E, et al. A survey of payload-based traffic classification approaches[J]. IEEE Communications Surveys & Tutorials, 2013, 16(2): 1135-1156.

[7] Xu C, Chen S, Su J, et al. A survey on regular expression matching for deep packet inspection: Applications, algorithms, and hardware platforms[J]. IEEE Communications Surveys & Tutorials, 2016, 18(4): 2991-3029.

[8] Wang, W., Zhu, M., Zeng, X., Ye, X., & Sheng, Y. (2017, January). Malware traffic classification using convolutional neural network for representation learning. In 2017 International conference on information networking (ICOIN) (pp. 712-717). IEEE.

[9] Gao M, Ma L, Liu H, et al. Malicious network traffic detection based on deep neural networks and association analysis[J]. Sensors, 2020, 20(5): 1452.

[10] Fu, C., Li, Q., Shen, M., & Xu, K. (2021, November). Realtime robust malicious traffic detection via frequency domain analysis. In Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (pp. 3431-3446).

[11] Li C, Wang J, Ye X. Using a recurrent neural network and restricted Boltzmann machines for malicious traffic detection[J]. NeuroQuantology, 2018, 16(5).

[12] He M, Wang X, Zhou J, et al. Deep-feature-based autoencoder network for few-shot malicious traffic detection[J]. Security and Communication Networks, 2021, 2021.

[13] Yang J, Liang G, Li B, et al. A deep-learning-and reinforcement-learning-based system for encrypted network malicious traffic detection[J]. Electronics Letters, 2021, 57(9): 363-365.

[14] Foulds J, Frank E. A review of multi-instance learning assumptions[J]. The knowledge engineering review, 2010, 25(1): 1-25.

[15] Sultani, W., Chen, C., & Shah, M. (2018). Real-world anomaly detection in surveillance videos. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 6479-6488).

[16] Zaheer, M. Z., Mahmood, A., Astrid, M., & Lee, S. I. (2020, August). Claws: Clustering assisted weakly supervised learning with normalcy suppression for anomalous event detection. In European Conference on Computer Vision (pp. 358-376). Springer, Cham.

[17] Cances, L., Pellegrini, T., & Guyot, P. (2018, November). Sound event detection from weak annotations: weighted-gru versus multi-instance-learning. In Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)-IEEE AASP 2018 (pp. 64-68). Tampere University of Technology.

[18] Carbonneau M A, Granger E, Raymond A J, et al. Robust multiple-instance learning ensembles using random subspace instance selection[J]. Pattern recognition, 2016, 58: 83-99.

[19] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

[20] Pang G, Cao L, Chen L. Homophily outlier detection in non-IID categorical data[J]. Data Mining and Knowledge Discovery, 2021, 35(4): 1163-1224.

[21] Lashkari, A. H., Kadir, A. F. A., Taheri, L., & Ghorbani, A. A. (2018, October). Toward developing a systematic approach to generate benchmark android malware datasets and classification. In 2018 International Carnahan Conference on Security Technology (ICCST) (pp. 1-7). IEEE.

[22] Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009, July). A detailed analysis of the KDD CUP 99 data set. In 2009 IEEE symposium on computational intelligence for security and defense applications (pp. 1-6). Ieee.