A stacking-based classification approach to android malware using host-level encrypted traffic

Zhixing Xue¹, Weina Niu*¹, Xixuan Ren¹, Jie Li¹, Xiaosong Zhang¹ and Ruidong Chen¹

¹ School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

*Corresponding author’s e-mail: niuweina1@126.com

Abstract. In recent years, smartphones have been developing fast. Android, a mobile platform convenient and open to all, has attracted more audience than any one of its counterpart. However, mobile devices are frequently attacked by malware, which calls to malware detection. Currently, we are lacking studies of Android malware detection based on ensemble learning. In this work, we propose a model to detect Android malware. The model takes the encrypted traffic that the malware generates as input. Through clustering, the model removes the third-party traffic and retains the purity of the first-party traffic. The model extracts traffic features to construct host-level traffic fingerprint and classifies the malware through stacking-based ensemble learning. We use the publicly available dataset CICAndMal2017 to build the classification model. This dataset successfully classifies malware into different categories. In the controlled experiments we use SVM and Random Forest models. The results show that our model is significantly more accurate in classifying malware than SVM and Random Forest models, with an accurate rate of 96.7% in the optimal condition.

1. Introduction

With the emergence of the mobile Internet, mobile devices have become part of our life. According to Ericsson [1], the number of mobile devices worldwide was 8 billion by the end of 2020 and is expected to be 8.8 billion by the end of 2026. The most popular mobile operating system, Android, accounted for 72.83% of the market in 2020 [2]. We enjoy the convenient life the Internet and smartphones bring us, but we also suffer from the trouble of malware attacks. According to the official report published by 360 Security [3], For the whole year of 2020, 360 Mobile Security Research Office intercepted about 4,564,000 mobile malicious samples, a 151.3% increase compared to that in 2019 (1,809,000). Android malware are of many types, such as ransomware, phishing software, adware, etc. The malware risk users’ privacy, rights, interests, and even their normal life. Therefore, malware detection has become an important issue.

In recent years plenty of researches have been conducted on malware detection, among which anomaly network traffic detection [4] as a new method is favored by researchers because of its convenience in data collection and high accuracy in malware detection. At present, there are four main methods of Android malware detection [5]: port-based methods, DPI-based methods, statistics-based methods, and behavior-based methods. In recent years, the application of advanced confusion technology prevails, which makes port-based detection less effective. The vast majority of traffic is transmitted in the way of encryption transmission, so deep packet inspection is not applicable. To cope
with the obstacles mentioned above, we now detect malware mainly by statistical-based detection and behavior-based detection.

That one single machine learning model results in low accuracy malware classification is addressed as below: we use a malware detection method based on the combination of statistics and behavior and classify malware by a stacking-based ensemble learning method. This work contributes in the following aspects:

- To eliminate the adverse effects on the classification results brought by the third-party traffic, we use clustering to clean the source dataset. Since we have removed the third-party traffic, the classification accuracy is significantly improved.
- We extract different features from the traffic, investigate their impact on classification results, and map flow-level features to host-level traffic fingerprints with statistical methods. In this way, we have achieved the transformation from flow-level features to host-level features.
- We classify malware by stacking-based ensemble learning method. The classification model fully combines the advantages of various machine learning methods, and the classification accuracy reaches 96.7%.

The rest of this work is organized as follows. In Section 2, we discuss the characteristics of Android malware and existing approaches to detect malware using network traffic. In Section 3, we introduce the computational method of our proposed model. Section 4 is the report of the experiments and analysis of the results. And Section 5 is a conclusion of this work.

2. Related work

With the development of the Internet and mobile devices, more and more types of malware emerge. Modern malware are often characterized by different code obfuscations and repackaging technologies, making many traditional static analysis methods ineffective. However, the research [6] pointed out that most malware requires a network connection to interact with C&C servers to upload payloads or accept commands. By studying access applications of malware, Yerima et al. [7] found that more than 93% of the malware required a network connection to realize malicious behavior.

Recently, an increasing number of researchers are using network traffic to detect malware. Ilund et al. [8] proposed a lightweight approach where the authors detect Android malware by identifying malicious behavior in network traffic. However, this method cannot be applied to encrypted traffic. In 2017, Arora et al. [9] proposed a feature ranking algorithm by analyzing features in network traffic, which reduces the number of features to be analyzed and shortens the time of processing. Lashkari et al. [10] proposed a new Android malware detection model based on network transmission characteristics. The authors extracted nine features from the network traffic, with which they successful classified Android software into benign software, malware, and adware.

Currently, artificial intelligence is developing rapidly, and many researchers apply machine learning and deep learning methods to malware classification. In 2017, Tan et al. [11] used the Bayesian model to classify malware by collecting its features in the execution of Android software. Wang et al. [12] proposed a new traffic classification method. The authors transformed the raw network traffic into image fingerprints and then used convolutional neural networks to classify the traffic. Here they use deep learning to replace manual extraction of features. In 2018, Tarar et al. [13] proposed an opcode-based Android software analysis method. They used multiple machine learning methods to classify malware, significantly improving the classification accuracy.

However, stacking-based ensemble learning method to classify malware is rarely used, nor do researchers pay attention to removing the third-party traffic from the original traffic. Similarly, the current detection method is mainly for the analysis of flow-level traffic characteristics and ignore the host-level traffic characteristics. In this work, our model first removes the third-party traffic from the original traffic and converts the flow-level traffic features into the host-level traffic features. Stacking-based ensemble learning method is used to detect malware, which significantly improves detection accuracy. Apart from malware identification, our model also classifies malware into specific categories.
3. Methodology

In this section, we will describe the specific approach of our model. Figure 1 shows the design of our Android malware detection system. In the data pre-processing stage, the flow in the PCAP file is clustered to filter out third-party traffic. Then, remove them from the original data to get pure first-party traffic. After that, the statistical features of the flow are extracted from the traffic, and statistical methods are applied to convert the features of the flow into features of the host-level traffic. Finally, the ensemble learning method is used to classify malware.

![Figure 1. The design of the proposed Android malware detection system.](image)

3.1. Third-party traffic removal

In our experiments, we found that not all traffic generated by malware can perform negative behavior, so we divided the traffic into two categories. The first type is the first-party traffic, which often communicates with the server set up by malware developers to carry out malicious behavior of a greater degree of harm. The malicious behavior of varies widely among different malware and has distinctive characteristics which is therefore often used as the critical standard for classification. The second type is the third-party traffic, which is usually the public traffic generated by software in its operation, such as third-party services using, and public URLs access. This kind of traffic exists in both malware and benign software. In Section 4.1, we do a comparison experiment to see whether the remove of the third-party traffic is effective. It turns out that removing third-party traffic improves the accuracy of classification results.

In this work, the method of removing third-party traffic is divided into two steps. The first step is to filter out possible third-party traffic preliminarily by clustering. We first extracted the flows in the PCAP file and randomized the flows in different PCAP files. Since the data packets in each flow have duplicate five tuples, we distributed all the flows evenly into several areas and cluster the flows within each area. If a flow in an area is shared by more than one PCAP, we considered this flow as possible third-party traffic and mark this flow.

Because of the random nature of area selection, the single judgement that a flow is shared by multiple PCAP files is insufficient to indicate that the flow is a third-party traffic. So, in the second step, we need to count the percentage of the occurrence of this flow in all areas. We set a threshold \( K \). If the proportion of occurrence of this flow in all areas exceeds the threshold \( K \), we marked the flow as the third-party traffic, and removed the flow from the original PCAP file; otherwise, the flow is treated as a first-party flow and not processed.

3.2. Host-level flow characteristics generation

Generally, not all flows generated during malware execution are malicious traffic, so we cannot mark all flows in the PCAP file generated by malware as malicious, we can only mark this PCAP file as
malicious. For this reason, we investigated a method to transform the features of the flow in the PCAP file into the features of the PCAP file.

In this work, we used Wireshark to extract flows from PCAP files. We also used CICFlowMeter tool [14] to extract 86 statistical features of streams, including the duration of flows, the time interval between flows, etc. Through experiments we removed the source IP, the source port, and other features that degrade the classification accuracy. In this way, we got 73 statistical features that are most suitable for classification.

We divided all the flows in each PCAP file into N groups and calculated the aggregation function for each feature value in each group with statistical methods. Seven aggregation functions such as mean, standard deviation, and median are included. Then the feature vectors flows of each group are combined to generate an N*7*73 feature vector to represent the entire PCAP file.

3.3. Ensemble learning method based on stacking
In this work, we employed stacking-based ensemble learning model as the malware classification model and machine learning classification algorithm SVM and bagging-based ensemble learning algorithm, Random Forest, as the control experiment. We employed scikit-learn’s GridSearch to tune all the models so that the parameters of the model are optimal under the condition provided by the dataset.

4. Experiment and evaluation
In this section, we show the results of our experiments and evaluate them. We use scikit-learn as the machine learning framework, and all our investigations are conducted on Windows 10 (64-bit) with an i5-10400, 2.90Ghz processor, and 16GB of RAM.

4.1. Datasets
In this work, we chose CICAndMal2017 [10] as the training and testing dataset. This publicly available dataset is from the Canadian Institute for Cybersecurity Research. Through the cellphones in use, it collects traffic data of 1700 benign software and 426 malware from the google play market. All the traffic data are stored as PCAP files, and the malware are classified into 4 categories, namely Adware, Ransomware, Scareware and SMSware. Data used in our experiment covers the whole dataset, as it is shown in Table 1.

|                | Benign software | Adware | Ransomware | Scareware | SMSware |
|----------------|-----------------|--------|------------|-----------|---------|
| APKs           | 1700            | 91     | 111        | 112       | 109     |
| Size           | 18.5G           | 8.43G  | 3.32G      | 5.05G     | 2.22G   |
| Flows          | 1240874         | 387620 | 384370     | 437878    | 266242  |

We chose 75% of the data as the training set and 25% of the data as the test set. The experiments are cross-validated. The average accuracy is for the measurement of the experimental model.

4.2. Third-party traffic removal
We first carried out experiments on the effectiveness of removing third-party traffic. We used the SVM, Random Forest, and Stacking models for our experiments, taking the values of N as 5, 10, 15, and 20 in order, and setting the number of features of the flow to 73. Experiments were carried out in environments without and with the removal of the third-party traffic, respectively. The influence of different thresholds K on the accuracy of classification results was tested, and the value of threshold K was finally set to 0.6. The experimental results are shown in Figure 2, and it can be found that the removal of the third-party traffic has different degrees of impact on the accuracy of all the three models. The accuracy of SVM is increased by 8% -10%, the accuracy of Random Forest is increased by 7% -8%, and the accuracy of Stacking is increased by 5% -7%. Therefore, it can prove that removing the third-party traffic is effective for our experiments.
4.3. Feature selection and host-level traffic feature generation

Our experiments investigated different features. We observed the accuracy of the model by randomly selecting a certain number of features from 86 statistical features provided by CICFlowMeter. And we filtered the features that make the model less accurate by controlling variates, as it is shown in Figure 3. Our experiments indicate that when the number of statistical features is less than 70, the model accuracy generally tends to increase as the number of features increases. In contrast, when the number of features exceeds 70, the accuracy hardly grows or grows negatively, so we cannot get the highest accuracy by selecting all the 86 features. Through further experiments, we removed 13 features that can degrade accuracy and kept 73 features as the final flow features.

![Figure 2. The experiment of removing third-party traffic.](image1)

![Figure 3. The experiment of feature selection.](image2)

We also conducted experiments to study the effect of picking different N values on the accuracy rate. Similarly, we conducted experiments on SVM, Random Forest, and Stacking models, respectively. In the experimental process, tune was made when training the Stacking-based ensemble learning model. The final model structure is as follows: in the first layer Gaussian Bayesian, Random Forest, Adaboost, and LGBM classifiers are employed, respectively, and the final classifier selects logistic regression. The experimental results are shown in Figure 4. We can see that the accuracy of the SVM classifier without ensemble learning is the lowest, between 80% and 90%. The accuracy of the Random Forest ensemble learning method based on Bagging is slightly higher than that of the SVM model, while the Stacking-based ensemble learning method after tuning the parameters has significantly higher accuracy than Random Forest and SVM, with the accuracy generally above 90%. The accuracy is the highest when N = 14, which is 96.7%.

![Figure 4. The experiment of different values of N.](image3)
4.4. Performance comparison
Some of the machine learning methods and deep learning methods were employed to detect malware, which were then evaluated in the same dataset, CICAndMal2017. The evaluation results are shown in Table 2. The ‘B/M’ column represents whether this method is for binary classification of malware or multi-classification of malware. When the value of this column is B, it means that this method can only classify Android software as malware and benign software. When the value of this column is M, it means that this method can not only classify Android software into malware and benign software but also determine the specific category of malware. The ‘Method’ column represents the classification method. ML represents machine learning, and DL represents deep learning.

| Work       | Pub time | Accuracy | Method | B/M |
|------------|----------|----------|--------|-----|
| This paper | N/A      | 96.7%    | ML     | M   |
| [14]       | 2017     | 79.1%    | ML     | B   |
| [12]       | 2017     | 88.2%    | DL     | M   |
| [15]       | 2017     | 93.9%    | DL     | M   |
| [16]       | 2020     | 80.0%    | ML     | M   |
| [17]       | 2020     | 95.2%    | ML     | M   |

5. Conclusion
This work proposed a model for detecting and classifying Android malware through the features of encrypted traffic and the ensemble learning method based on Stacking. The model first removed the third-party traffic from the traffic by clustering to obtain pure first-party traffic. The model then extracted 86 flow-level features through feature engineering and selected 73 statistical features that are most suitable for classification through comparative experiments. Then the flow-level feature vectors were merged into host-level traffic feature fingerprints using statistical methods. Finally, the model classified malware using ensemble learning method based on stacking. The experimental results show that the removal of third-party traffic and the selection of appropriate traffic features significantly improve the classification accuracy up to 96.7%. Although this method achieves good performance in detecting Android malware, it is limited by the samples in the dataset and can not well detect new types of malware. So our future task is to collect and analyze more malware samples, and try to establish different classifiers for different types of malware.

Acknowledgments
This work was partially supported by the National Natural Science Foundation of China (Grant no. 61902262), the National Defense Innovation Special Zone Program of Science and Technology (Grant no. JG2019055), the Special Central Finance project (Grant Y030202059018060).

References
[1] Erisson. (2020) Erisson Mobility Report, November 2020. https://www.ericsson.com/en/mobility-report/reports.
[2] StatCounter Global Stats. (2021) Mobile Operating System Market Share Worldwide - July 2021. https://gs.statcounter.com/os-market-share/mobile/worldwide/2021.
[3] 360 Internet Security Center. (2021) 2020 mobile phone security report. https://zt.360.cn/1101061855.php?did=1101061451&did=610755858.
[4] Ahmed, M., Mahmood, A.N., Hu, J. (2016) A survey of network anomaly detection techniques. Network and Computer Applications., 60: 19–31.
[5] Biersick, E., Callegari, C., Matijasevic, M. (2013) Data traffic monitoring and analysis. Springer-Verlag, Berlin.
[6] Zhou, Y., Jiang, X. (2012) Dissecting Android Malware: Characterization and Evolution. In: 2012 IEEE Symposium on Security and Privacy. San Francisco. pp. 95-109.

[7] Yerima, Suleiman Y., Sezer, S., McWilliams, G. (2012) Analysis of Bayesian classification-based approaches for Android malware detection. IET Information Security. San Francisco. 8: 25-36.

[8] Ilund, D., Pucher, A., Schaub, T. (2011) Detecting Android Malware on Network Level. University of California., 12.

[9] Arora, A., Peddoju, S.K. (2017) Minimizing Network Traffic Features for Android Mobile Malware Detection. In: 18th International Conference on Distributed Computing and Networking. Hyderabad. pp. 1-10.

[10] Lashkari, A.H., Kadir, A.F.A., Gonzalez, H., Mbah, K.F., Ghorbani, A.A. (2017) Towards a Network-Based Framework for Android Malware Detection and Characterization. In: 2017 15th Annual Conference on Privacy, Security and Trust. Calgarys. pp. 233-23309.

[11] Tan, M., Yu, M., Wang, Y., Li, S., Liu, C. (2017) Android malware detection combining feature correlation and Bayes classification model. In: 2017 9th IEEE International Conference on Communication Software and Networks. GuangZhou. pp. 664-668.

[12] Wang, W., Zhu, M., Zeng, X., Ye, X., Sheng, Y. (2017) Malware Traffic Classification Using Convolutional Neural Network for Representation Learning. In: 2017 International Conference on Information Networking. DaNang. pp. 712-717.

[13] Tarar, N., Sharma, S., Krishna, C.R. (2018) Analysis and Classification of Android Malware using Machine Learning Algorithms. In: 2018 3rd International Conference on Inventive Computation Technologies. XiaMen. pp. 738-743.

[14] Samara, M., El-Alfy, E.M. (2019) Benchmarking Open-Source Android Malware Detection Tools. In: 019 2nd IEEE Middle East and North Africa COMMunications Conference. Manama. pp. 1-6.

[15] Wang, W., Zhu, M., Wang, J., Zeng, X., Yang, Z. (2017) End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In: 2017 IEEE International Conference on Intelligence and Security Informatics. Beijing. pp. 43-48.

[16] Abuthawabeh, M., Mahmoud, K. (2020) Enhanced Android Malware Detection and Family Classification, using Conversation-level Network Traffic Features. The International Arab Journal of Information Technology., 17: 607–614.

[17] Zhou, J., Niu, W., Zhang, X., Peng, Y., Wu, H., Hu, T. (2020) Android Malware Classification Approach Based on Host-Level Encrypted Traffic Shaping. In: 2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing. ChengDu. pp. 246-249.