Fine-grained Android Malware Detection based on Deep Learning

Dongfang Li[1,3], Zhaoguo Wang[2,3] and Yibo Xue[1,3*]
[1]Department of Computer Science and Technology, Tsinghua University, Beijing, China
[2]School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
[3]Tsinghua National Lab for Information Sci. & Tech., Tsinghua University, Beijing, China
{lidf15, wangzhaoguo}@mails.tsinghua.edu.cn, yiboxue@tsinghua.edu.cn

***Abstract*—Android smartphone users have been suffering from the security problems these years. There is a serious threat to the network security and privacy brought by the mobile malware. In this paper, we use the deep-learning-based method to detect Android malware and implement an automatic detection engine to detect the families of malicious applications. The results of the evaluation show that the engine can detect 97% of the malware at 0.1% false positive rate (FPR) when detecting the fine-grained malware families.**

***Keywords*—Android Malware Detection; Deep Neural Network; Fine-grained Classification; Smartphones Security**

I. Introduction

The smartphones have been an indispensable part of modern human's life and as one of the most popular operating system, the market share of Android Smart Phone is increasing year by year. Unfortunately, there are so many malicious applications troubling users due to the openness nature of the Android operating system. These malicious applications from Google Play or other third-party markets often seriously threaten users' privacy and the device's security. Thus, there are urgent needs to stop the spread of the Android malware to create a safe environment for Android smartphone users.

Detection based on rule and signature is one of the most common approaches to detect malware. However, the signature-based method does not work efficiently with the rapid development of the malware due to its strong reliance on the fingerprint database[1]. In view of such conditions, the learning-based methods are proposed to detect malware from benign applications automatically. As one of new field of machine learning, deep learning can also be applied in malicious applications detection[2][3]. These methods only give the results whether applications are malicious or not, lacking in the reason why they are malicious and what types they are on earth.

To make up for these problems, we aim to detect Android malware using the deep neural network. We extract comprehensive static features from applications and use these datasets to train the detection models that can classify the fine-grained malware families when detecting malicious applications. Evaluation shows that it outperforms the related learning-based models with a 97.16% detection accuracy at 0.1% false positive rate when detecting the detailed malware families.

II. Fine-grained Deep Neural Network

As suffering from so many types of different malware, we should focus on further understanding the specific types of malware and its characteristics rather than only the result whether applications are malicious or not. Thus based on the deep neural network model, we introduce the fine-grained deep neural network for malware family detection.

The first step to characterize the Android applications is to get static features from given applications. In particular, we focus on the Manifest.xml file and the classes.dex file of applications, where the permissions required, API called and other information about the applications can be obtained. Then the deep learning model is trained using backpropagation, which means minimizing the value of loss function by the gradient descent method. Consider we have L = {0,1,2,3…} as the layer in the neural network, $\boldsymbol{y}^{(L-1)}$ as the input value for the layer L, $\boldsymbol{y}^{(L)}$ as the output value of the layer L and $W^{(L)}$ as the weights of the layer that converts the input values into output values. Besides, $F^{(L)}$ is the corresponding activation function and $b^{(L)}$ is the bias for the model. In addition, r is independent samples from Bernoulli distribution. It is obvious that the input data is the origin feature vectors extracted from Android applications when L=0. Thus we can construct the deep neural network as follow

$$\boldsymbol{y}^{(L)} = F(\, W^{(L)} \,(\, \boldsymbol{y}^{(L-1)} \bullet r^{(L)}) + b^{(L)} \,) \tag{1}$$

Note that the • means pointwise vector product. For hidden layers in our network, we choose PReLU function as the activation function because it is proposed for its high efficiency and dynamical adjustment. Let $\mathbf{z}$ be the input of F, δ_i is one of the n-dimensional output values and the activation function of each neuron can be described as (2)

$$F\,(\, \mathbf{z}^{(L)} \,) = (\, \delta_1^{(L)} , \ldots, \delta_i^{(L)}, \ldots, \delta_n^{(L)} \,) \tag{2}$$

Based on the deep neural network, the fine-grained model introduces the Softmax units. In particular, let n be the number of the neurons in output layer and the activation function for fine-grained output layer is

Table 1 Classification accuracies with different deep learning model constructions

Number of neurons	*Precision*	*Recall*	*F1*	Number of neurons	*Precision*	*Recall*	*F1*
[256,256,256,256,256,256]	**96.94%**	**95.23%**	**96.08%**	**[1024,1024,1024]**	**97.16%**	**91.23%**	**94.10%**
[256,256,256,256,256]	**97.15%**	**94.18%**	**95.64%**	**[1024,1024,512]**	**96.63%**	**92.09%**	**94.30%**
[256,256,256,256]	**96.34%**	**90.28%**	**93.21%**	**[1024,1024,256]**	**96.23%**	**92.47%**	**94.31%**
[256,256]	**94.28%**	**87.99%**	**91.03%**	**[512,512,512]**	**95.79%**	**93.33%**	**94.54%**
[256]	**92.34%**	**81.60%**	**86.64%**	**[256,256,256]**	**95.07%**	**93.80%**	**94.43%**

$$\delta(i)^{*} = e^{z(i)}\left(\sum_{j=1}^{n} e^{z(i)}\right)^{-1} \tag{3}$$

where $\delta(i)^{*}$ is the i-th neuron of the output layer, that is, it means the probability about whether the applications are the corresponding types to the malware families that the i-th neurons stand for. Note that all the $\delta(i)^{*}$ belong to (0,1) and the normalization factor guarantees that the sum of all the $\delta(i)^{*}$ is 1, which means the model is logical. What's more, we use Logarithmic likelihood loss function as the *loss* function to speed up the training process and make it more precise.

The Softmax method extends the binary classification model based on the deep neural network to multi-label, which makes it possible to detect fine-grained Android malware families.

III. Evaluation

To evaluate the performance of detection malware of our engine, we introduce a public data set [4]. There are 131,611 applications including 5,560 malware samples and 123,453 benign samples. To make the best use of the data set, we evaluate the engine as following.

There are several parameters need to be set to train the deep neural networks, among which the most important are the numbers of layers and the number of neurons in every single layer. These two main parameters decide the structure of the deep neural networks and we tried different pairs of the two key factors to find the collocation with better performance. The results are shown in Table 1.We can see from Table 1 that the model can achieve a 97.16% precision when the number of layers is 3 and each layer has 1024 neurons, whose performance is better than any other model with other structure.

Moreover, we evaluate the performance of the fine-grained deep neural network classifier using the ROC(Receiver Operating Characteristic) Curve as shown in Fig.1. We select some typical Android malware families and show the detection performance of their classifiers. Note that we use the ROC curve to describe the performance of each classifier and show the macro-average ROC curve of all the malware families. We can see from Fig.1 that almost all the curve can achieve a 0.1% FPR when the macro AUC(Area under the Curve of ROC) is 0.9977, which we think is good enough to detect the Android malicious applications and show their families in detailed.

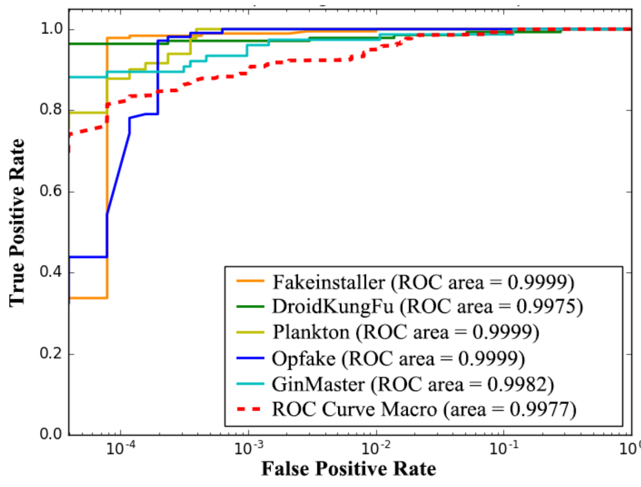

Fig. 1 ROC Curve of fine-grained classifier.

IV. Conclusion

In this paper, We implement a fine-grained malware detection engine based on deep neural network to detect malicious applications. The engine shows the detail malware families and other information rather than only the result whether the applications are malicious, which makes it more convincing. We conduct the evaluation experiments with dataset from real world and results show that the engine perform well. In the future, dynamic features of Android applications and more types of samples will be considered.

Acknowledgment

This work is supported by the National Key Technology R&D Program of China under Grant No. 2015BAK34B00.

References

[1] Z. Fang, W. Han, and Y. Li, Permission based Android security: Issues and countermeasures, Computers & Security, vol. 43, pp. 205–218, 2014.

[2] Yuan Z, Lu Y, Wang Z, et al. Droid-Sec: deep learning in android malware detection[C]//ACM SIGCOMM Computer Communication Review. ACM, 2014, 44(4): 371-372.

[3] Martinelli F, Mercaldo F, Saracino A. BRIDEMAID: An Hybrid Tool for Accurate Detection of Android Malware[C]//Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security. ACM, 2017: 899-901.

[4] Arp D, Spreitzenbarth M, Hubner M, et al. DREBIN: Effective and Explainable Detection of Android Malware in Your Pocket[C]//NDSS. 2014.