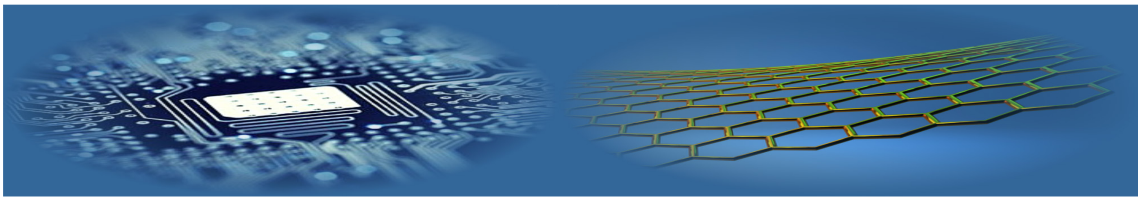

Using convolutional neural network for Android malware detection

Isil Karabey Aksakalli

Erzurum Technical University, Faculty of Engineering and Architecture, Department of Computer Engineering, ERZURUM

**Corresponding author's e-mail: isil.karabey@erzurum.edu.tr*

Received 30 March 2019, www.cmnt.lv

Abstract

With the increasing usage of smart mobile devices, the number of applications developed for these devices is already increasing day by day. Nearly all functionalities (sending e-mails, searching the internet, messaging via internet, making bank account transactions etc.) performed by using computer are carried out on mobile devices anymore. However, misuse of personal information emerges through malicious applications in the devices and these applications render the devices unusable. In the literature and industry, new methodologies have been proposed for mobile malware detection; however, there is still a research challenge to identify malwares on mobile applications and take precautions. In this paper, a permission-based model is implemented to detection of malware applications in mobile devices which have Android operating system. Permission-based features have been extracted from the apk files in the AndroTracker1 data set which is previously created in the literature. The results of classification techniques have been evaluated by applying four types of machine learning techniques (Support Vector Machine, k-Nearest Neighbor, Back Propagation) and these techniques have been compared with Convolutional Neural Network. The experimental results show that the permission-based model is highly successful using both machine learning technique and deep learning in the AndroTracker data set. Back Propagation gives the best result among the other machine learning techniques by 96.1% acurracy rate. Also Convolutional Neural Network has achieved success rate of 96.71%. This demonstrates that the accuracy rates of CNN and classical machine learning techniques close to each other and they have high accuracy rate because of small number of targets which are benign and malware.

Key words

Android, permission-based malware detection, convolutional neural network, machine learning

1 Introduction

Nowadays, computers have been replaced by more portable devices such as wristbands, smart mobile devices, tablets etc. with the advancement of the technology. Android Operation System (OS) is among the most popular OS' used in these devices. There are millions of applications on the Android operating system, and users can easily upload their applications to their mobile devices via the Android market. Although the mobile devices make life easier, malicious software developers try to access personal information through these apps that make life easier. They can access to users' devices by injecting malware such as virus, trojan into an apk file which is an extension of Android-based applications.

In this study, AndroTracker (http://ocslab.hksecurity.net/andro-tracker) dataset is used to identify whether Android based applications are malicious or not. The apk files are passed through the ApkReader tool and the application permissions are extracted from these files. Using this permission information, the data set is arranged as the learning algorithm can handle. In our study, Convolutional Neural Network (CNN) is used as a learning algorithm. Besides, the performance of the CNN algorithm is compared to some popular machine learning algorithms namely k-Nearest Neighbor (k-NN), Naive Bayes (NB), Back Propagation (BP) and Support Vector Machine (SVM).

The rest of this paper is structured as follows: Section 2 reviews the techniques used in malware detection field. Section 3 explains methods used in this work. Section 4 describes our evaluation dataset. Section 5 outlines the experimental results and discussions. Finally, the last section provides our conclusions.

2 Malware Analysis Techniques

Software or programs that display illegal behavior and used for malicious purposes are called malware. Mobile users download applications over unauthorized resources and malware included in these sources leaks the user's personal information. Since users are not aware of the features of applications such as permission information, API calls, intents and commands, they download malicious applications even in situations that require user permission. In order to protect from these software, various approaches are presented in the literature. These approaches are divided into three parts: static, dynamic and hybrid analysis.

2.1 STATİC ANALYSİS

In static analysis method [2 - 4], when the application is not at run time, the source code and binaries [7] of the application are examined and the static properties extracted from the application are used, such as the desired permissions and API calls in order to detect malware. Some of these methods are generally not resistant to obfuscation [9].

2.2 DYNAMIC ANALYSİS

In this method called behavioral analysis [5 - 6] analyzed software must be running while monitoring and following the behaviour of the application in an isolated environment [7]. This method is more resistant to obfuscation than static analysis. The main advantage of the method is to record the application behaviors and to determine the dynamic code load in the runtime process. On the other hand, dynamic methods offer limited scalability as they require additional cost in terms of processing and memory [9]. Therefore, it is a more difficult method to implement according to static analysis [8]. Due to this disadvantage of dynamic analysis, in this study, a permission based model is proposed using static analysis method applied to apk files.

Since both methods have advantages and disadvantages, hybrid analysis methods [10 - 12] using the advantages of these methods are also recommended in the literature.

2.3 HYBRID ANALYSİS

In this analysis method, the static and dynamic features of the application are used together to determine the suspicious behavior at runtime of the application and also the static bytecode tools are used in the analysis routines [13]. In static analysis part, features (API calls, permission data) are generally extracted from manifest files. In dynamic analysis part; file access and operations, receiver records, executable commands, content solving queries, dynamic suspicious calls, network operations etc. can be monitored actively via tools used at runtime of the application.

3 Method

Android OS based devices have applications with apk extension. Besides some parameters such as application versions, package name etc. specified by the developers along with some information such as camera access, SMS sending, microphone access photo album, vibration, internet access, etc. are operated depending on the user's permission. This information is stored in AndroidManifest.xml in each apk file and is prepared by the developer.

Permissions are reported when downloading applications from the market, but users usually download the application that they want to use by giving approval because they do not care the permission data or do not have any information about these permissions. This situation cause installing malware to device and to access the personal information with the consent of the user.

Determining whether apk files are malicious during operation cannot prevent malware from infecting the device. Therefore, in this study, malware detection is performed automatically by static analysis method without running the application.

The proposed system model is shown in Figure 1. In the system model, the apk files included in AndroTracker dataset are passed through an ApkReader open source library and the permission data from these files are extracted. This library accesses the AndroidManifest.xml file in each file and lists the required permissions for these files. This permission data is made into the features of malicious or benign applications to create a new data set. When the feature value is transferred to the data set, the permission data requested by the application is assigned to 1, and those that are not requested are assigned to 0. Then, machine learning techniques and CNN are applied to the new data set consisting of feature vectors.

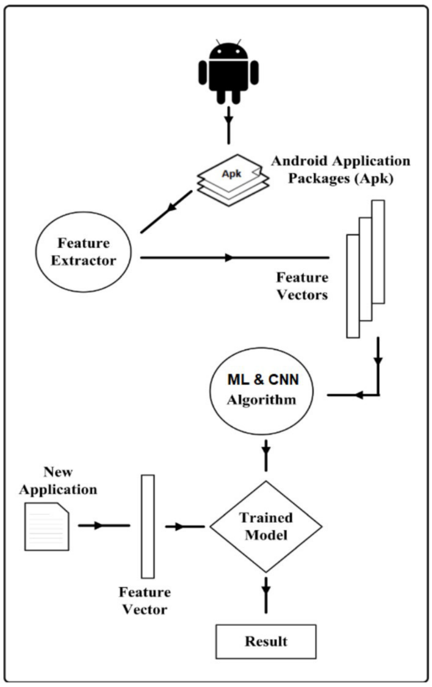

FIGURE 1 System model

3.1 EXTRACTING PERMISSION FEATURES

To detect whether raw dataset named AndroTracker includes malware or not, AndroidManifest.xml files included in the apk file need to be extracted. Thanks to the ApkTool, apk files in the raw data set, AndroidManifest.xml files are extracted and a new dataset is created with the .arff extension. There are 8353 sample and 493 permission data in the data set. Some of these data are WRITE_EXTERNAL_STORAGE, INTERNET, VIBRATE, CALL_PHONE, BILLING, SEND_SMS, CAMERA etc. If an apk file contains this permission data, 1 is assigned to the attribute in columns, if it does not contain this specified permission data, then 0 is assigned. After converting this data set which consists of binary values into csv file, machine learning and CNN algorithms are applied to the data set using various Python Spyder. A small part of the generated data set is shown in Table 1.

3.2 CLASSIFICATION

In this study, Convolutional Neural Network which is very popular in data classification in recent years and some successful machine learning algorithms (KNN), Naive Bayes (NB), Random Forest (RF) and Back Propagation) are compared to each other in terms of accuracy rates. These algorithms are briefly summarized as follows: k-Nearest Neighbor (kNN): The nearest neighborhood method (kNN) is a basic method used in the classification process. To classify a sample taken from the test set, the distance between each sample in the training set is calculated. Then, the sample with the number of k shortest distance from train set is selected, and the most common target in these examples is determined as the target of the test sample.

Naive Bayes (NB) Classifier: In Naive Bayes method, the probability of finding the target of the signal received from an unknown test data is obtained by multiplying the probability of finding the result of all the factors affecting the result. That means, the target of the class is C and the F values are signal strengths;

TABLE 1 A Small part of Generated dataset

Internet	SendSMS	Change WiFi State	Camera	Billing	Target
1	1	1	0	1	malware
0	0	0	1	0	benign
1	0	0	1	0	benign
1	0	0	0	0	benign
1	0	0	0	1	malware

$$P(C|F1,F2,F3,\ldots FN)=(\mathrm{P(C)}(PF1,F2,F3,\ldots FN|\mathrm{C}))/(\mathrm{P}\ (F1,F2,F3,\ldots FN)) \qquad (1)$$

the probability of the class of unknown signal belongs to C class is found using the calculation (1).

If the number of the class target is higher than the probability calculations, the class target of the test data is also determined as that label.

Support Vector Machine (SVM): The purpose of support vector machines is to obtain the optimum accuracy of the targets of the test data, which can be separated as accurately as possible.

Random Forest (RF): The Random Forest method, which is known as the collective classification method, uses the method of "voting" with the estimates obtained from many decision trees for the classification operation. These decision trees are independent of each other and consist of variables selected from the training set. Among the randomly selected variables in the training set, the best information (entropy) is used as a distinctive variable. In the class prediction of unknown targeted data there are as many predictions as the decision trees developed and the most rated class by voting method is determined as the target of the new data.

Back Propagation (BP): In the back propagation method, the signals belonging to the training set are the input values of the artificial neural network and the class targets are the output values. After training the system by taking into account the weight of the learning coefficient and the training data, the test is performed for signals with unknown target.

Convolutional Neural Network (CNN): Convolutional neural network (CNN) based on artificial neural network, is one of the specialized architectures of deep learning. The difference between artificial neural networks and CNN is that CNN can handle many more hidden layers than ANN. When the number of layers in artificial neural networks is more than 5, the system becomes complicated and slows down, and the situation becomes incurable. In the Deep learning method which has many nodes and parameters, many architectures including Concentration Boltzmann Machines, automatic coders, and Convolutional Neural Network are used. Since the Convolutional Neural Networks used in the image are one of the most popular deep learning methods, this method is also evaluated on two dimensional signal data. In the convolutional neural networks consisting of several layers that can be trained, each layer has three layers, including the feature pooling layer, the filter bank layer, and the non-linear layer [14].

The architecture created for malware detection of the convolutional neural network is given in Figure 2.

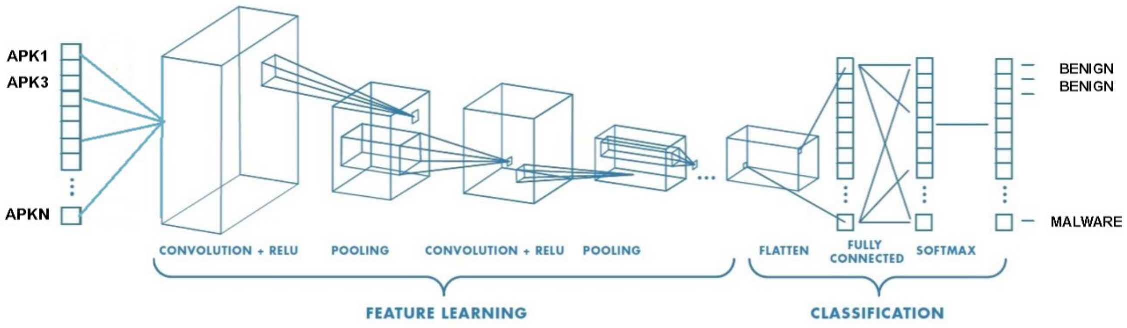

FIGURE 2 Designed CNN network using APK permissions [15]

3.3 EVALUATION

In order to calculate the accuracy of the statistical model that performs the classification on the known target data, 10-fold cross validation method is applied. In this technique, the data set is divided into 10 parts instead of separating the training and test sets in specific proportions.Each of these parts is used as test data and the remaining nine parts are used as training data. Each part can be evaluated as test data since the parts change in each iteration. This method has more advantages than other evaluation metrics since all data is used as both train and test data and the results are obtained by taking the average of 10 accuracy rates. An instance of 4-fold cross validation is explained in Figure 3.

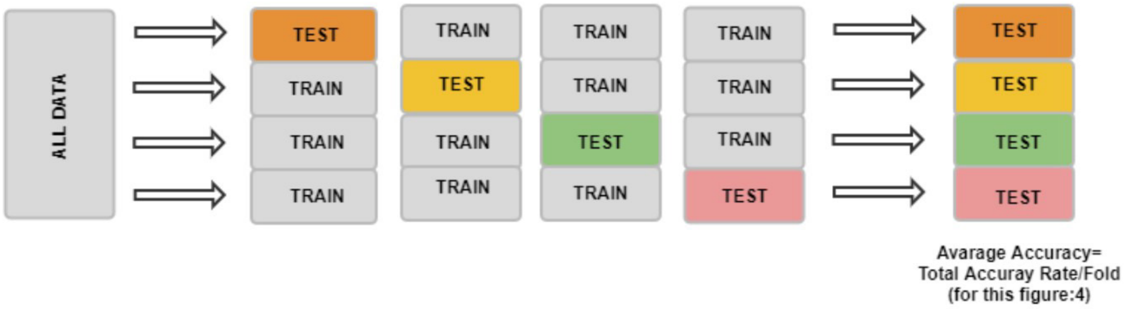

FIGURE 3 An example of n-fold cross validation model

4 Dataset

There are many studies which are evaluated on different datasets for different purposes for android malware detection. Some of them are composed of only malware samples or various malware families, whereas some of them contain only benign samples. We also have a handicap in this field that new malware samples arise constantly. In this paper, therefore, AndroTracker dataset is selected as evaluation material, because it includes 51179 benign and 4554 malware applications from different malware families [1]. However, it is obtained a subset of AndroTracker by randomly selecting 5K samples both in benign and malicious categories. In this process, it is aimed to create a small and balanced dataset as the size of the dataset is out of the scope of this paper. In decompilation process, however, a challenge has emerged that compelled us to remove some application files from which any manifest file or permission information could not be extracted. This is because the reverse engineering tool (https://ibotpeaches.github.io/Apktool/) that is used fails in some cases. After the reverse engineering process, therefore, the AndroTracker dataset just includes 3933, and 4421 samples in benign and malicious categories respectively.

5 Experimental Results

In this section, 8353 apk files are determined whether they are malware or not through classifiers. Also 494 permission data extracted from AndroTracker dataset. The data set is separated by 10-fold cross validation method including training and test data. In this method, the data set is divided into 10 parts and each piece is tested in order. The accuracy of the classifier is determined by calculating the average accuracy rate in this process which repeats 10 times. In this study, four different popular machine learning methods (k-NN, NB, BP, SVM, RF) and Convolutional Neural Network (CNN) are used to classify the dataset. The default parameter values of these classifiers are described in Table 2.

TABLE 2 Default Parameters for Classification Techniques

Method	Default Value
kNN	k=5 distance function= euclidean distance

Support Vector Machine	c=1.0
	kernel=PolyKernel
	tolerance=0.001
	ε= 1.0E-12

Back-propagation	hidden layer= a ((attributes + classes) / 2) learning rate=0.3 momentum=0.2

CNN	# of layer=3
	# of epochs=1000
	optimizer=adam
	activation=relu
	loss=categorical_crossentropy

The accuracy percentages obtained from classification methods using the default parameters shown in Table 2 are given in Table 3. When the results obtained, it is seen that the success of all classification methods is more than 90%. However, considering the working times of the algorithms, it is observed that machine learning methods classify in longer time than deep learning, especially in the BP algorithm (approximately one and a half hours).

TABLE 3 The Accuracy Rate of The Classification Techniques

Classifier	Accuracy Rate
kNN	95.8%
SVM	95.9%
NB	91.2%
BP	96.1%
CNN	96.71%

Since there are only two targets in the data set, it is obtained high accuracy rate which is around 90% in both machine learning algorithms and CNN algorithm. However, it is seen in Table 4 that CNN algorithm gives more than 95% accuracy rate even if test data is more than train data (0.60, 0.80 etc.). With this aspect, CNN is differentiated from machine learning methods.

TABLE 4 The accuracy rate of the CNN algorithm using different validation splits

Validation split	Accuracy Rate (%)
0.20	96.9
0.40	96.8
0.60	96.1
0.80	95

6 Conclusions

In Android malware analysis using the static analysis method, it is proved with the experimental result that the permission based model works effectively both in machine learning methods and in deep learning. However, while the accuracy of these methods is very close to each other, some differences were observed in terms of running times of algorithms. Although the BP algorithm using the artificial neural network works very slowly, it is observed that convolutional neural network, which performs BP (back propagation) method for each layer through multiple intermediate layers, provides a much faster and more accurate rate than all other algorithms. In addition, machine learning methods were classified using 80% training, 20% test data, even in convolutional neural network (epoch), and vice versa (20% training, 80% test) accuracy was found to be around 90%. Since the number of data used in the study is high and the dataset has only two targets consist of malicious and benign, it is observed that the level of accuracy is 96.71%.

References

[1] Kang H J, Jang J, Mohaisen A, Kim H K 2014 AndroTracker: Creator Information based Android Malware Classification System *The 15th International Workshop on Information Security Applications (WISA)*

[2] Schmeelk S, Yang J, Aho A 2015 Android Malware Static Analysis Techniques *CISR '15 Proceedings of the 10th Annual Cyber and Information Security Research Conference*

[3] Kang H, Jang J, Mohaisen A, Kim H 2015 *Detecting and Classifying Android Malware Using Static Analysis along with Creator Information*

[4] Karabey I, Coban O 2016 Mobile Malware Detection using Classification Techniques *Proceedings of Academics World 20th International Conference, Istanbul*

[5] Shabtai A, Kanonov U, Elovici Y, Glezer C, Weiss Y 2012 "Andromaly": A Behavioral Malware Detection Framework for android devices *Journal of Intelligent Information Systems* **38**(1) 161-90

[6] Schütte J, Fedler R, Titze D 2015 ConDroid: Targeted Dynamic Analysis of Android Applications *IEEE 29th International Conference on Advanced Information Networking and Applications*

[7] Milosevic N, Dehghantanha A, Choo K R 2017 Machine learning aided Android malware classification *Computers and Electrical Engineering* **61** 266-74

[8] Sawle P D, Gadicha A B 2014 Analysis of Malware Detection Techniques in Android *International Journal of Computer Science and Mobile Computing* **3** 176-82

[9] Karbab E B, Debbabi M, Derhab A, Mouheb D 2018 MalDozer: Automatic framework for android malware detection using deep learning *Digital Investigation* **24** 48-59

[10] Yuan Z, Lu Y, Xue Y Droiddetector: android malware characterization and detection using deep learning *Tsinghua Science and Technology*

[11] Vidas T, Tan J, Nahata J, et al. 2014 A5: automated analysis of adversarial android applications *Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices*

[12] Yang F, Zhuang Y, Wang J 2017 Android Malware Detection Using Hybrid Analysis and Machine Learning Technique *International Conference on Cloud Computing and Security* 565-75

[13] Ali-Gombe A I, Saltaformaggio B, Xu D, Richard III, Golden G 2018 Toward a more dependable hybrid analysis of android malware using aspect-oriented programming *Elsevier* **73** 235-48

[14] Cengil E, Cinar A A New Approach for Image Classification: Convolutional Neural Network *European Journal of Technic* 96-103

[15] **E-source:** https://www.mathworks.com/solutions/deep-learning/convolutional-neural-network.html (Accessed 10 February 2019)

AUTHORS

Isil Karabey Aksakalli, 23 September 1990, Erzurum, Turkey

Current position, grades: Research Assistant
University studies: Erzurum Technical University, Erzurum, Turkey; Computer Engineering
Scientific interest: Machine learning, Deep Learning, Indoor localization, Optimization Algorithms,Microservices architecture
Publications: Author of 15 conference papers
Experience: 5 years of researching