Indonesian Journal of Electrical Engineering and Computer Science
Vol. 4, No. 1, October 2016, pp. 240 ~ 244
DOI: 10.11591/ijeecs.v4.i1.pp240-244

Android Malware Detection Using Backpropagation Neural Network

Fais Al Huda, Wayan Firdaus Mahmudy, Herman Tolle
Master Program, Faculty of Computer Science, Brawijaya University, Jl Veteran Malang, Indonesia
Corresponding author, e-mail: faisalhuda@outlook.com, wayanfm@ub.ac.id, emang@ub.ac.id

Abstract

The rapid growing adoption of android operating system around the world affects the growth of malware that attacks this platform. One possible solution to overcome the threat of malware is building a comprehensive system to detect existing malware. This paper proposes multilayer perceptron artificial neural network trained with backpropagation algorithm to determine an application is malware or non-malware application which is often called benign application. The parameters that used in this study based on the list of permissions in the manifest file, the battery rating based on permission, and the size of the application file. Final weights obtained in the training phase will be used in mobile applications for malware detection. The experimental results show that the proposed method for detection of malware on android is effective. The effectiveness is demonstrated by the results of the accuracy of the system developed in this study is relatively high to recognize existing malware samples.

Keywords: *android manifest, classification, smartphone, static analysis*

1. Introduction

The rapid growth of smartphones development also has the negative impact on the growth of malware that attacks this platform. This condition is shown on the McAfee report that the amount of malware attack in 2015 has exceeded nine million attacks per year, especially on the android operating system from Google.Android devices had been sold up to 1 billion units[1]. and the number of applications on the AppStore has exceeded 1 million applications [2]. So do not be surprised if this operating system became the target of various types of malware from the malware that only affects the performance of the battery until the malware is very dangerous as trying to steal valuable information.

The amount of malware that has been very large regarding quantity and variants can no longer be ignored, in particular with the phenomenon of bringing your own device (BYOD) that have occurred in all parts of the world makes the detection of malware on mobile platforms is a priority. With the increasing number of employees who bring instruments themselves to the office although improving the performance of the employee but also open security risk seriously enough. It is possible that the malware can steal important information from the employee's personal data or relevant information from the company.

Malware is an application that can harm the user due to its ability to interfere with the performance of the system or could damage the security of the system. A malware application can steal important personal data from the user such as image data or the user's credit card information. Even a malware can reduce the performance of the system by doing much service behind the scenes resulting the increase of memory cost and processor usage that makes the battery drops drastically.

Several studies related to the malware have been done [3-5]. However, most of these studies only focused on free applications on the appstore third parties and did not target specific categories. Moreover, no one does research related to BYOD phenomenon is now increasing in various companies and other institutions.

In this paper, a method for detecting the malware on mobile devices through static analysis using a backpropagation MLP algorithm is proposed. The backpropagation MLP is used because it has a good concept to recognize a complex pattern [6] that is expected to identify with both an existing malware on mobile devices.

Received May 3, 2016; Revised August 26, 2016; Accepted September 11, 2016

2. Research Method

In the Android operating system, all applications that have the important features that need to use the android API must obtain prior permission from the user in the form of permissions list of things to be agreed at the time of initial app installation. Android operating system will only allow the application to run the API by the list of permits at the time of the initial approval. However, security model like this has a very big weakness that could interfere with the performance and safety of the system because the application can run an activity behind the scenes without the knowledge of the user.

2.1. Multilayer Neural Network

Multilayer neural networks are often called a multilayer perceptron (MLP) is one of the models that exist on artificial neural networks. In the network model, there are three main layers; there are the input layer (input layer), a hidden layer (hidden layer), and the output layer (output layer), the structure of this network model can be seen in Figure 1

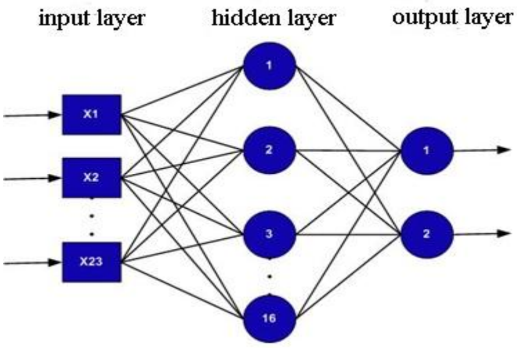

Figure 1. Multilayer artificial neural network

Given this, the hidden layer network model can solve cases that are more complex than single-layer neural network models that can only resolve cases that are linear.

2.2. Backpropagation Algorithm

Backpropagation algorithm is an algorithm that is most often used to train the MLP and is one of the learning algorithms in the category of supervised learning, this category of output results will be compared with the targeted value. In the process of training to use this algorithm, there are several stages that passed that signal the beginning of a unit of input will be heading to hidden units and resumes to the unit output of all tracks have respective weights. When the output of the network is not in accordance with the desired target, then it will make a step back (propagation) to update the weights that exist in the input unit and hidden unit to get more accurate solution [7].

Network training is repeatedly done until the network obtained the minimum error rate or the maximum iterations have been reached. For error rate using the theory of Mean Square Error (MSE). In general, the smaller the MSE level then the better the network architecture. MSE calculation can be seen in Equation (1).

$$\text{MSE} = \frac{1}{n}\sum_{i=1}(t_i - y_i) \qquad (1)$$

n = amount of data
t_i = the target value
y_i = the output value

In the process of updating the weights are known term rate of learning is often called the learning rate, this is used as a value to adjust how much weight value will be changed, the greater the value of learning rate, the greater the change in the weight value. The range of values used in the learning rate between 0 and 1.

3. Implementation

In the implementation phase is divided into several stages. The first stage is starting from the stage of extraction features to look for features that are appropriate because the data can not be directly used. The next stage of designing the architecture of the MLP, and the last stage is the implementation of the backpropagation algorithm on a system that has been designed.

3.1. Feature Extraction

Malware data that is used as a sample obtained from contagiodump.blogspot.com [8]. Retrieval features that will be used as input parameters of the system obtained from the manifest file that will be extracted from the APK file are the installation file on the Android operating system. In this file, there is a list of permissions that are used by the application. The hallmark of malware that attackedAndroid operating system is the number of permissions used in the app exceeds the actual needs of normal applications similar to those applications [9]. The use of batteries is also included as a parameter input, to measure the use of the battery based on the use of the internet and GPS permission [10]. And the last feature that is used as a parameter is the size of the APK file to be analyzed. For a list of permissions based parameters in this study using 20 permissions most common use is in the malware application [11].

Table 1. List of the most common permission in the malware application

Permission name	Explanation
INTERNET	Permission to open network sockets / internet access
ACCESS_COARSE_LOCATION	Permission to access approximate location of the phone tower or Wi-Fi
VIBRATE	Permission to access the vibrator
WRITE_EXTERNAL_STORAGE	Permission to write to external storage
READ_SMS	Permission to read the sms messages
WRITE_SMS	Permission to write messages sms
READ_CONTACTS	Permission to read the contact list
BLUETOOTH	Permission to connect to a paired device Bluetooth
WRITE_CONTACTS	Permission to write the contact
DISABLE_KEYGUARD	Permission to turn off the keyguard
WAKE_LOCK	Permission to use Power Manager to keep the processor from sleep states
RECORD_AUDIO	Permission for audio recording
ACCESS_FINE_LOCATION	Permission to access precise location using GPS, tower, and Wi-Fi
ACCESS_NETWORK_STATE	Permission to access information about networks
READ_PHONE_STATE	Permission to access the state of the phone
SET_ORIENTATION	Permission to set your screen orientation
CHANGE_WIFI_STATE	Permission to change the state Wifi connection
READ_LOGS	Permission to read data log system
BLUETOOTH_ADMIN	Permission to discover and pair with Bluetooth devices
RECEIVE_BOOT_COMPLETED	Permission to determine whether the system has finished booting

3.2. MLP Architecture Design

MLP architecture used in this study consisted of 23 neurons in the input layer, 16 neurons in the hidden layer, and one neuron in the output layer. Neurons in the input layer obtained from permission list of 20 most frequently used. Represented a value of 1 if the permissions are used are found in the manifest file and the value 0 if permission is not found. two neurons representing battery usage represented the value {1,1} if the battery usage included in weight category, the value {1.0} if use of battery in the medium category, and the value {0,0} if categorized as low category, and one neuron latter represents the size of the APK file. To enter the APK file size first be normalized so that the size of all files to be trained to fall in the range of values from 0 to 1 so that it can meet the sigmoid activation function. The formula for normalization is shown in Equation (2).

$$\text{normalized value} = \left(\frac{previous\ value - a}{b - a} \text{ x } 0.8\right) + 0.1 \qquad (2)$$

previous value = before normalized value
a = the smallest data value
b = the biggest data value

4. Result and Discussion

The accuracy of the algorithms should be evaluated. There are several stages of testing which start from testing the number of iterations, testing the value of learning rate, the number of neurons in the hidden layer, and the final test is the comparison of training data and test data. The first phase is testing the number of iteration; this test is useful for finding the optimal number of iterations in the algorithm training. The results of these tests are shown in Figure 2.

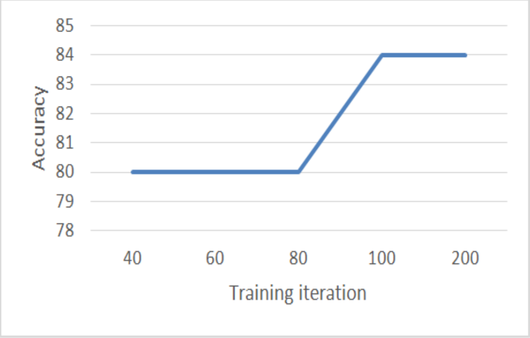

Figure 2. Test based on training iteration

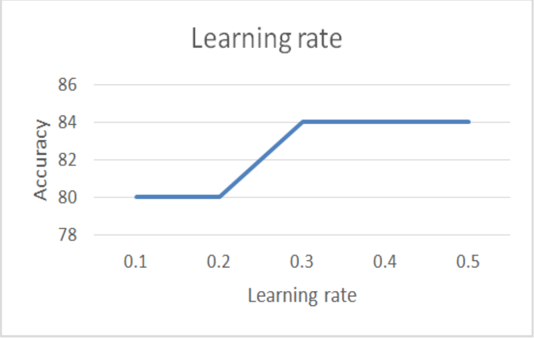

Figure 3. Test based on learning rate

As shown in Table 2, after reaching the 100th iteration accuracy can not increase. Therefore it can be concluded that the value of this iteration was sufficient to be used.The next test is an experiment on learning rate value because the value is quite a big influence on the training of artificial neural network model. The test results for these parameters are shown in Figure 3.

From the results of experiments on learning rate is known that the learning rate value of 0.3 to 0.5 generates the same accuracy values so as to the next training use the value of learning rate 0.3. The next test was conducted on a parameter number of nodes in the hidden layer, the number of nodes in the hidden layer is also a considerable influence on the training of artificial neural network model. The number of nodes also affects the level of computing on systems that are built, the more the number of nodes that exist on this layer will increase the training time algorithm model. Results of testing the number of nodes in the hidden layer are shown in Figure 4.

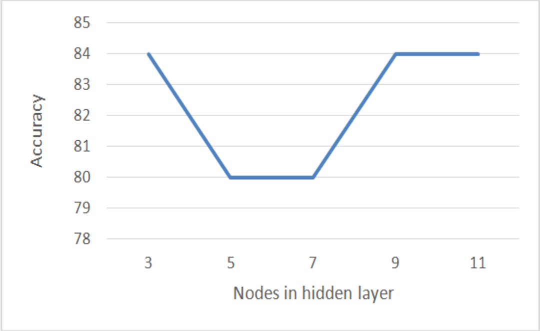

Figure 4. Test based on the total amount of node in hidden layer

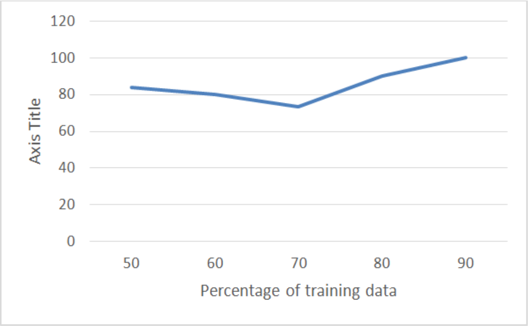

Figure 5. Test based on comparison of training data and testing data

As shown in Figure 4 that the number of nodes in the hidden layer 3, 9, and 11 receive the same results, but when considering the level of computing, the number of 3 nodes can be considered as the best because of the training time will be faster than others.

The final experiment is the comparison of training data and test data. This result is shown in Figure 5.

Experiments on the amount of training data show a decrease in accuracy when the number of training data by 60% and 70%, but when the amount of 80% or more training data showed an increase in accuracy of up to 100% when the amount of training data by 90% of the total data. This is because the amount of test data that is too small, so the variation of the data is not too much.

5. Conclusion

Based on the results of the implementation is carried out to the 50 data patterns extracted from an android manifest file which contains permissions list of the app, the level of battery usage and the size of the apk file obtained. ANN architecture is 23-16-2 (23 units of neurons in the input layer, 16 neurons in the hidden layer unit, and two units of the target the output layer). The number of iterations applied only 100 iterations because increasing the number of iterations is not changing the accuracy from the system significantly because the limited number of malware samples used in this study. Based on the comparison of training data and test data by using comparisons 0.9 for training data and 0.1 for testing data,100% accuracy is achieved.However, it possible because of the limited sample data used. In the near future, experiment with a large number of the data sample and added another static analysis parameter or using the combined analysis (hybrid) is a potential research topic.

References

[1] ZDNet. Report: Google Play Store has more apps than Apple App Store. http://www.zdnet.com/article/report-google-play-store-no-has-more-apps-than-apple-app-store/. 2015.

[2] CNet. Android shipments in 2014 exceed 1 billion for first time. http://www.cnet.com/news/android-shipments-exceed-1-billion-for-first-time-in-2014/. 2015]

[3] Burguera I, Zurutuza U, Nadjm-Tehrani S*. Crowdroid: behavior-based malware detection system for Android.* SPSM’11, ACM. 2011.

[4] Felt AP, Finifter M, Chin E, Wagner D. *A survey of mobile malware in the wild*. In ACM workshop on security and privacy in mobile devices(SPSM), ACM. 2011: 3-14.

[5] Grace M, Zhou Y, Wang Z, Jiang X. *Systematic detection of capability leaks in stock android smartphones.* NDSS’12. 2012.

[6] Yajin Z, Xuxian J. Dissecting android malware: Characterization and Evolution. San Francisco, CA. 2012.

[7] Basu JK, Bhattacharyya D, Kim T, Use of artificial neural network in pattern recognition. *International Journal of Software Engineering and Its Application.* 2010.

[8] Huang H. A hybrid neural network prediction model of air ticket sales. *TELKOMNIKA*. 2013; 11(11): 6413-6419.

[9] Mobile malware mini dump. contagiodump.blogspot.com/2011/03/take-sample-leave-sample-mobile-malware.html. 2015.

[10] Aung Z, Zaw W. Permission-based android malware detection. *IJSTR.* 2013.

[11] Koundel D, Ithape S, Khobaragade V. Malware classification using Naïve Bayes classifier for android OS. *IJES.* 2014.

[12] Moonsamy V, Rong J, Liu S. Mining permission patterns for contrasting clean and malicious android applications. *Future Generation Computer Systems.* 2014; 36: 122-132.