VFAST Transactions on Computer Sciences
http://vfast.org/journals/index.php/VTCS@ 2016 ISSN(e): 2308-8168; ISSN(p): 2411-6335
Volume 4, Number 1, January-December, 2016

pp. 31-35

CLASSIFICATION OF ANDROID MALWARE APPLICATIONS USING FEATURE SELECTION AND CLASSIFICATIN ALGORITHMS

ALTYEB ALTAHER [1]

` [1]Faculty of Computing and Information Technology, King Abdulaziz University, Rabigh -21911,Saudi Arabia
Email: aaataha@kau.edu.sa

Revised May 2016

ABSTRACT. *Smartphones have become a potential part of our lives, and this led to a continues increase in the number of smartphone users. The growing number of users attracts hackers to develop malware applications to steal the private information and causing potential financial losses. Due to the fast modifications in the technologies used by malware developers, there is an urgent need for more advanced techniques for malware detection. In this paper, we propose an approach for Android malware classification based on features selection and classififcation algorithms. The proposed approach uses the permissions used in the Android app as features, to differentiate between the malware apps and goodware apps. The information gain algorithm is used to select the most significant permissions, then the classification algorithms NaivBayes, Random Forest and J48 used to classify the Android apps as goodware or malware apps. The experimental results show that random forest algorithm achieved the highest precision of 0.898 with lowest false positive rate of 0.110.*

Keywords*:* Android ; Malware detection; Classification algorithms.

1. **Introduction.** The global market for smartphones is growing continuously. According to Gartner [1], the international sales of smartphones reached 403 million smartphones in the fourth quarter of year 2015, which is 9.7 percent rise over the equivalent period in the previous year 2014. Android is the most dominant operating system for smartphones, representing about 85% of the international smartphone market. The flexibility of installing smartphone applications from different application markets is the main characteristic of smartphones. The principal applications markets (e.g., Google Play and App Store) contain more than one million applications Accessible and can be downloaded either by users free or after payment, the number of installed applications in the global approximately is more than 100 billion mobile device applications. The huge number of smartphone applications in many life's aspects leads to an exponential increase of malware applications. Kaspersky [2], reported that the amount of malware targeting the smartphones increased more than three times in 2015, when compared to 2014. The potentially dangerous threats in 2015 were ransomware, malware able to get unlimited rights on the compromised smartphone, and data stealers, including financial malware. Also Kaspersky [2] stated that the total number of new malwares detected in 2015 is 884,774 new malicious programs, and this number is a a three-fold increase on 2014 (295, 539).

2. Related Work. To secure the Android smartphones, several research studies focused on Android malware detection,. One of the Common approaches for malware detection are the signature-based approaches, which extract the signatures features from malware applications. Although it is good for the detection of malware with known signatures, it is not suitable for detecting malware with unknown signatures. Kirin is an example for signature-based Android malware detection technique [3]. The application's permissions are used as signatures in Kirin, and Kirin decides whether the application matches a particular signature by exploring its

manifest file, the main disadvantage of Kirin is that it generates many false negatives and false positives. Another two examples for signature based approaches are Stowaway [4] and RiskRanker [5].

Another approach for Android malware detection is the behavioral detection technique described in [6]. behavioue based approaches collect information that describe the behvaiour of Android app to be used as features, to classify the apps as malware or goodware. TaintDroid [7], DroidRanger [8] and DroidScope [9] are approaches that can observe the behavior of applications during its running. Although the behavior based detection approaches are efficient in detecting malware app, monitoring the apps during its running generates high overhead on the smartphones.

Classification approaches have been proposed to classify the Android apps as malware or goodware apps. Shabtai et al. [15] used feature selection algorithms such as information gain , Chi-Square and Fisher Score to select the important features of the Android app, they used the classification algorithms : BN, Decision Tree, Histogram, K-means and Logistic regression to differentiate between the malware and goodware apps. They achieved 89% of accuracy when classifying Android tools and games apps. Sanz et al. [16] used the permissions in Android app to classify malware apps. They Used dataset contains 239 Android malware apps, and the classification algorithms Random forest, Naïve bayes as classifiers. The random forest algorithm achieved the best classification accuracy of 86.41%. Aswini et al. [17] used the permissions in the Android app's manifest file as features for the classification of malware apps. Their results show that the Random Forest achieved the highest classification accuracy of 87%. In our previouse work [18], we used a neuro-fuzzy classifier to classify the Android malware apps based on Android permissions, the best classification accuracy obtained by the proposed classifer was 75%.

3.The proposed approach for Android malware classification. The proposed approach consists of two steps. The first step involves the extraction and selection of the most significant features that can help in the discrimination between the malware and goodware. The second step uses the classification algorithms for the classification of Android malware.

3.1 Used Dataset

We used malware dataset from Genome project [10], the dataset contains 1,200 malware apps collected in the period between August 2010 and October 2011. The dataset is made available by researchers to assist the researchers in the security field to find effective malware detection approaches [11]. the Genome dataset consists of malware apps only. To evaluate the performance of the classification algorithms in terms of malware detection , we added goodware apps to the Genome dataset, by downloading goodware apps from Google Play. Our dataset consists of 100 malware app and 100 goodware app.

3.2Fetures Extraction and Selection.

We used open source tools like to extract the permissions from the application package (APK) file. Permissions are important components of Android application, permissions are used to control the application's access to the resources available in the smartphone . The used permissions in Android application should be declared in the application's Manifest.xml file. Table 1 shows examples for permissions and their uses.

Table 1 : Examples of Android Permissions

Permission	Usage
android.permission.INTERNET	Allows the application to connect the internet, it could be used by malware apps to send the user information to attackers , through the internet connection.
android.permission.CHANGE_CONFIGURATION	Allows the application to change configuration files of the smartphone.
android.permission.SEND_SMS	Allows the application to send SMS message .Malware apps can use this permission to send messages and causes financial losses to the smartphone user.
android.permission.CALL_PHONE	Allows the application to perform phone calls , malware apps can use this permisstion to do phone calls without user notification. The users will loss money for unwanted phone calls.

After the extraction of the permission features, we used the information Gain algorithm[13] to rank the permissions based on their importance for the classification as shown in Figure 2.

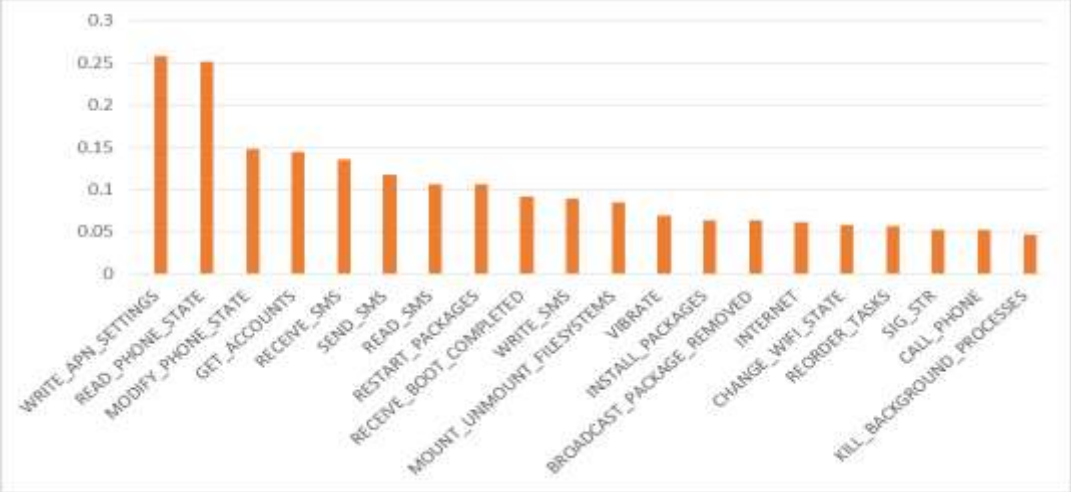

Figure1: Top 20 permissions ranked based on their importance for classification.

3.3 Using classification algorithms for malware classification. To perform the malware classification, we used three classification algorithms, Naïve Bayesian, Random Forest and J48.

The Naïve Bayesian algorithm is based on Bayes' theorem with independence assumptions between forecasters. Bayesian reasoning is used with decision making. It uses the prior knowledge about events for predicting the events in the future. Naïve Bayesian works without need for complex iterative parameter estimation, and that made it mainly most useful for huge datasets.

Random forest [12] is a collaboration of unpruned classification , Prediction is made by combining the predictions of the ensemble. Random forest commonly exhibits a considerable performance enhancement over the single tree classifier such C4.5.

J48 is to some extent modified C4.5 in WEKA. The C4.5 algorithm produces a classification-decision tree for the particular data set by recursive splitting of data. The decision is grown using Depth-first approach. The algorithm reflects the potential tests that can partition the data set and chooses a test that provides the greatest information gain. [14]

3.4 Performance Evaluation Metrics

To evaluate the performance of different classification algorithms , we used the following metrics:

1- Precision (P): represents the Android apps classified as malware that are really malware.

$$p = \frac{TP}{TP+FP} \quad (1)$$

2- Recall: It is also called as True positive rate (TPR). It is the rate of number of positive applications classified correctly.

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

False Positive Rate (FPR): It is number of goodware apps. misclassified as malware apps

$$FPR = \frac{FP}{FP+TN} \quad (3)$$

Where True Positives (TP): The number of malware apps classified as malware. True Negatives (TN): The number of goodware apps classified as goodware. False Positives (FP): The number of goodware apps classified as malware. False Negatives (FN): The number of malware apps classified as goodware.

Table 2 : The comparison between the three classification algorithms

Algorithm	TP Rate	FP Rate	Precision	Recall	ROC Area
NaivBayes	0.845	0.155	0.872	0.845	0.936
Random Forest	0.890	0.110	0.898	0.890	0.948
J48	0.870	0.130	0.870	0.870	0.926

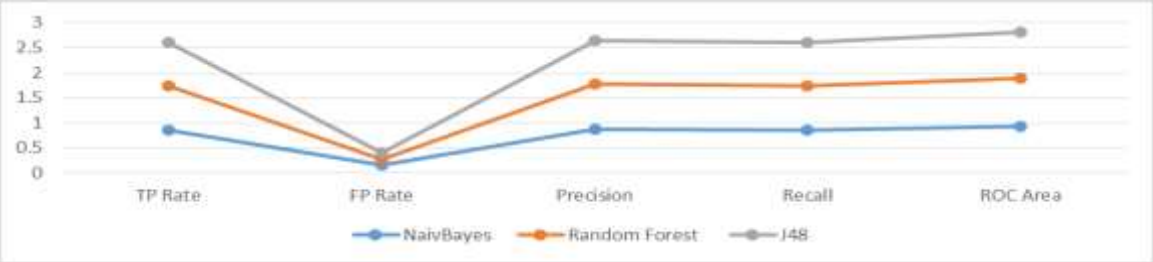

Figure 2: The performance of the three classification algorithms.

Figure 2 and Table 2 show the performance of the three classification algorithms, its clear form the table 2 that the random forest algorithm achieved the best performance in terms of the all used metrics. Approximately, the NaivBayes and J48 algorithms achieved the same precision. However, J48 algorithm achieved better TP and FP rates of 0.870 and 0.130 respectively.

Based on the malware classififcation accuracy achived by the Random Forest algorithm , the performance of our proposed approch is reasonable when compared with our previouse work[18]and other approches [15], [16] and [17] as shown in Table3.

Table 3: The performance of the proposed approach compared with other approchs

Classification approach	Pecision
The approach proposed in this paper	0.898
K-ANFIS [18]	75%
Droid Permission Miner [17]	87%
Puma [16]	86.41%.

5. Conclusion. Malware apps are real threats for smartphones, as they can steel private information and causes potential financial losses. The proposed approach uses the permissions used in the Android app as features, to differentiate between the malware apps and goodware apps. The information gain algorithm is used to select the most significant permissions, then the classification algorithms NaivBayes, Random Forest and J48 used to classify the Android apps as goodware or malware apps. The experimental results show that random forest algorithm achieved the highest precision of 0.898 with lowest false positive rate of 0.110. For future research, we will use more features to describe the Android app such as the Application Programming Interface (API) used in app.

REFERENCES

[1] Gartner, http://www.gartner.com/newsroom/id/3215217 Accessed 10/4/2016.

[2] Kaspersky, http://www.kaspersky.com/about/news/virus/2016/The_Volume_of_New_Mobile_Malware_Tripled_in_2015 Accessed 10/4/2016.

[3] Enck, W., Ongtang, M., & McDaniel, P. (2009). On lightweight mobile phone application certification. In *Proceedings of the 16th ACM conference on Computer and communications security* (pp. 235-245). ACM.

[4] Felt, A. P., Chin, E., Hanna, S., Song, D., & Wagner, D. (2011, October). Android permissions demystified. In Proceedings of the 18th ACM conference on Computer and communications security (pp. 627-638). ACM.

[5] Grace, M., Zhou, Y., Zhang, Q., Zou, S., & Jiang, X. (2012). Riskranker: scalable and accurate zero-day android malware detection. In Proceedings of the 10th international conference on Mobile systems, applications, and services (pp. 281-294). ACM.

[6] Bose, A., Hu, X., Shin, K. G., & Park, T. (2008). Behavioral detection of malware on mobile handsets. In Proceedings of the 6th international conference on Mobile systems, applications, and services (pp. 225-238). ACM.

[7] Enck, W., Gilbert, P., Han, S., Tendulkar, V., Chun, B. G., Cox, L. P., ... & Sheth, A. N. (2014). TaintDroid: an information-flow tracking system for realtime privacy monitoring on smartphones. *ACM Transactions on Computer Systems* (TOCS), 32(2), 5.

[8] Zhou, Y., Wang, Z., Zhou, W., & Jiang, X. (2012, February). Hey, You, Get Off of My Market: Detecting Malicious Apps in Official and Alternative Android Markets. In NDSS.

[9] Yan, L. K., & Yin, H. (2012). Droidscope: seamlessly reconstructing the os and dalvik semantic views for dynamic android malware analysis. *In Presented as part of the 21st USENIX Security Symposium (USENIX Security 12)* (pp. 569-584).

[10] Android Malware Genome Project, 2014. http://www.malgenomeproject.org/.

[11] Zhou, Y., & Jiang, X. (2012). Dissecting android malware: Characterization and evolution. In *Security and Privacy (SP), 2012 IEEE Symposium on* (pp. 95-109). IEEE.

[12] Breiman, L., Random forests, Machine Learning 45(1), 5-32, 2001.

[13] Mori, T. (2002). Information gain ratio as term weight: the case of summarization of ir results. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1* (pp. 1-7). Association for Computational Linguistics.

[14] Zhao, Y., & Zhang, Y. (2008). Comparison of decision tree methods for finding active objects. Advances in Space Research, 41(12), 1955-1959.

[15] 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), 161-190.

[16] Sanz, B., Santos, I., Laorden, C., Ugarte-Pedrero, X., Bringas, P. G., & Álvarez, G. (2013). Puma: Permission usage to detect malware in android. InInternational Joint Conference CISIS'12-ICEUTE´ 12-SOCO´ 12 Special Sessions (pp. 289-298). Springer Berlin Heidelberg.

[17] Aswini, A.M., Vinod, P.: Droid Permission Miner: Mining Prominent Permissions for Android Malware Analysis. In: 5th International Conference on the Applications of the Digital Information and Web Technologies (ICADIWT 2014), pp. 81–86 (2014).

[18] Abdulla, S., & Altaher, A. (2015). Intelligent Approach for Android Malware Detection. KSII Transactions on Internet and Information Systems (TIIS), 9(8), 2964-2983.