International Journal of Technical Innovation in Modern Engineering & Science (IJTIMES)
Impact Factor: 5.22 (SJIF-2017),e-ISSN:2455-2585
International Conference on
Recent Explorations in Science, Engineering And Technology (ICRESET'19)
Volume-5, Special Issue-March, 2019.

Comparative Analysis of Malware Android Apps Detection Using Machine Learning Approach

Gowri S PG Scholar, Dr.R.Bhavani Assistant Professor

Department of Computer Science and Engineering Government College of Technology, Coimbatore-641013

***Abstract*— Android is one of the most developed intelligent operating systems on mobile devices and has taken the most part of the cell phone market. The rapid evolution of mobile devices technology has increased the number of mobile malware in the application market, particularly when Android OS is widely adopted in the mobile devices. These Android malicious applications hidden behind the benign applications pose a serious threat to the Android platform. The end users and service providers are affected by malwares that are spreading around the world. In order to mitigate the threats posed by a malware app, there is a need for developing applications that detect Android malwares.**

This paper work focuses on the identification of Android malware using machine learning approaches. The objective of this paper is to classify the android application into benign or malware application. The proposed system utilizes the features of collected random samples of benign and malware apps to train the classifiers. The system extracts permissions used in the android applications as its features. With the extracted features, machine learning approaches are used to classify the applications as benign or malicious. To classify the android applications K-nearest neighbor, Decision trees, Support vector machines algorithms are used and the performance of the classifier is calculated.

Keywords—K-NN, Decision Tree, SVM, Classification

I. INTRODUCTION

Technologies have been changing rapidly in last decades. Nowadays almost everybody has a mobile phone. Recently it is more and more common that these phones have highly customizable software, capable of performing an enormous number of actions. The computational performance of a regular phone has increased so much, and the gap between the old ones and the new ones have become so wide that people started calling modern mobile phones as smart phones. Android is one of the most developed intelligent operating systems on mobile devices. It has taken the most part of the cell phone market.However the end users and service providers are affected by malwares that are spreading around the world. In order to mitigate the threats posed by a malware app, there is a need for developing applications that detect Android malwares. In this paper, a comparative analysis of various machine learning approaches on malware detection is carried out. The rest of the paper is organized in the following manner. Section II discuss the Background and related work, Section III presents the methodologies adopted in this work, Section IV presents the Experimental Results and in Section V the conclusion of the work is discussed.

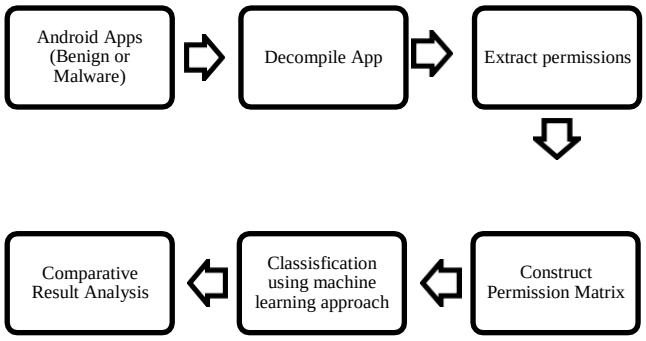
II. BACKGROUND AND RELATED WORKS

Smart phone users often neglect security issues, and directly confirm the pop-up windows without reading the permission requirement of the software. As a result, more number of smart phones have been implanted with virus. Many of the recent works focus on the permission based detection which analyses the androidmanifest file. Dong- Jie Wu, et al.,[1], proposed a mechanism which considers the static information including permissions, deployment of components, Intent messages passing and API calls for characterizing the Android applications behaviour. To improve the working of the classifier, K-means classifier is used along with K-nearest neighbours algorithm which has advantages over the others in terms of efficiency. R.Sato, in [2] proposed a light weight malware detection technique in which the system extracts the specific information described in the manifest file, and compares the extracted information with the keyword lists and then computes the malignancy score to judge the samples as malware or not. Chun-Ying Huang, et al.,[3], proposed a technique for classifying label-ing into three.

Site based labeling identifies apps as benign if it is downloaded from official Google market. Scanner based labeling depends on the anti-virus scanner. Mixed labeling is carried out on both scanner based and site based labels. To evaluate the performance Machine learning algorithms including AdaBoost, Naive Bayes, Decision Tree (C4.5), and Support Vector Machine are used. PUMA[4], a method for detecting malicious Android applications through machine learning techniques, where the performance of machine learning classifiers are evaluated using K-fold cross validation. Wang., Liu and Zhang[5], proposed a technique in which the permission-induced risk in Android apps on three levels is explored in a systematic manner. In[6], An android malware detection method based on android manifest file, a new feature vector is extracted from the Android Manifest file, which combines the permission information and the component information of the Android application. The chi-square test method is used to filter out permissions which have high relevancy with the sample category. The extracted feature vectors are used to train the classifier based on Naive Bayesian algorithm. Varma, et al.,[7], proposed a method which has mainly three stages. Firstly, the permission fields are extracted from the android manifest file of the apps. Second, a data base of all the permissions for both normal and malware data is established and finally the machine learning algorithms are used to classify and identify the malware in android applications. Koli, J. D.[8], have proposed a machine learning-based malware detection system for Android platform. The proposed system utilizes the features of collected random samples of goodware and malware apps to train the classifiers. The system extracts requested permissions, and uses them as features in various machine learning classifiers to build classification model.

III. METHODOLOGY

The flowchart of the proposed system is given in Figure I. FIGURE I: FRAMEWORK OF THE SYSTEM



The system takes the collection of benign and malware applications as input. In order to use machine learning algorithms for classification of android malware, first the permissions in all the apps are to be extracted. To extract the features, each application package (apk) file will be decompiled into their source code in the form of AndroidManifest.xml and java classes by using malware analysis tool called Apktool. Android apps must request permission to access sensitive user data (such as contacts and SMS), as well as certain system features (such as camera and internet). Depending on the task, the system might grant the permission automatically or might prompt the user to approve the request. The purpose of a permission is to protect the privacy of an Android user. The permissions used in the applications are extracted using the aapt (Android Asset Packaging Tool). With the extracted permissions, the permission matrix is constructed by placing a 1 if the permission is present in the Androidmanifest.XML file and a 0 if not. Finally it will be stored in a CSV file. Using the permission matrix the machine learning classifiers are trained to classify the android applications. With the trained classifiers, the test data are classified as benign or malware applications. Finally the performance of each machine learning classifiers is analyzed.

To classify the applications into benign or malware three algorithms are used namely K-Nearest Neighbour, Decision Tree and Support vector machine.

1) K-Nearest Neighbour:

N algorithm is a non-parametric statistical methods for categorization and regression. It classifies a test sample by measuring the distance between the training samples and test sample. K nearest samples are choosed and majority voting is used to predict the category of the sample.

2) Decision Tree:

Decision tree is a categorization model that recursively partitions the training data into a tree structure in which each leaf node corresponds to a class label and attributes are represented on the internal node of the tree.

3) Support Vector Machine:

SVM finds the best hyper plane to separate the data into two parts. The hyper plane always maximizes the margin between those two regions or classes. The margin is defined by the farthest distance between the samples of the two classes and computed based on the distances between the closest samples of both classes, which are called supporting vectors. Test samples are then mapped into the same space. Based on which sides of the hyper plane they fall on, test samples are predicted to belong to the corresponding categories.

IV. EXPERIMENTAL RESULTS

A. DATASET DESCRIPTION

In order to conduct extensive analysis on permission usage, we need to establish a well-labeled app set. In this experiment, a total 398 applications are collected [9], among them 199 are Benign apps and 199 are malware apps.

B. RESULT ANALYSIS

To evaluate the effectiveness of an algorithm, the confusion matrix shown in Table I is built.

TABLE I: CONFUSION MATRIX

	Predicted as Benign	**Predicted as Malware**
Actual Benign	True Positive	False Negative
Actual Malware	False Positive	True Negative

Here,

True Positive(TP): Number of correctly identified benign applications.

True Negative(TN): Number of correctly identified malware applications.

False Positive(FP): Number of cases were recorded as Malware in nature but got predicted as Benign.

False Negative(FN): Number of cases were recorded as Benign in nature but got predicted as Malware.

The effectiveness of the algorithm is measured using the metrics like Precision, Recall, F1 Measure, and Accuracy which is given in equation (1), (2),(3)and (4) respectively.

V. CONCLUSION

The Android malicious applications hidden behind the benign applications pose a serious threat to the android platform. So there is a need for the android malware detection application. In this paper, Comparative analysis of Android malware application detection using machine learning algorithm is proposed. To classify the apps into benign or malware, the permissions used in applications

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1\ Measure = \frac{2 * Precision * Recall}{Precision + Recall}$$

$$Accuracy = \frac{TP + TN}{Total\ Number\ of\ Observations}$$

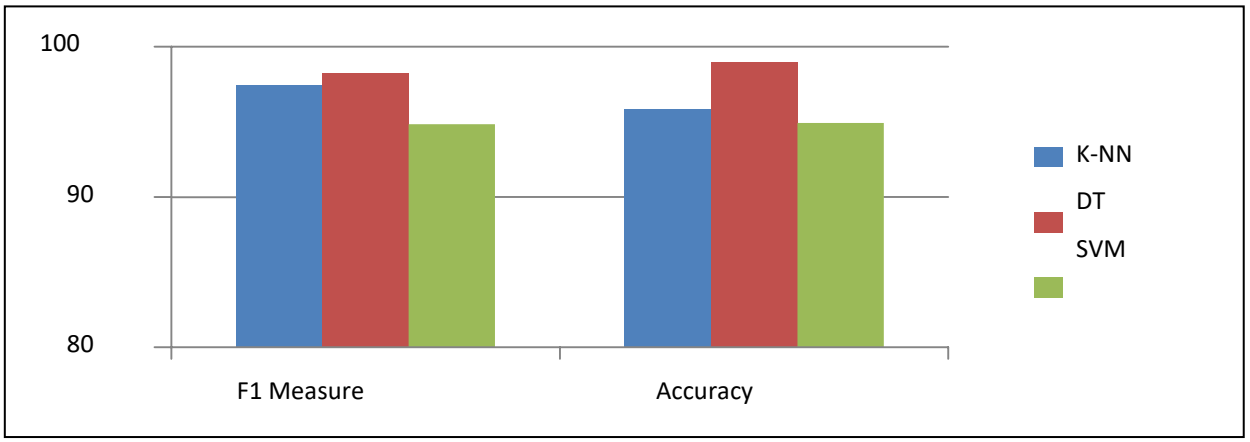
are used as a feature. In this paper K-nearest neighbor, Decision trees, Support vector machines is used to classify the android applications. From the results it is observed that Decision Tree classifier outperforms with the accuracy of 99.99% compared to K-NN and SVM which gives accuracy of 95.83% and 94.92%.

TABLE II: RESULTS ON TEST DATA

Performance Measures	**Classification Algorithms**		
	K-NN	**DT**	**SVM**
Precision	98.45	99.02	96.49
Recall	95.04	97.47	93.22
F1 Measure	97.47	98.24	94.83
Accuracy	95.83	98.95	94.92

FIGURE III: COMPARATIVE RESULTS

The experiment results are tabulated in Table II and depicted in Figure III. The classifier that outperforms in the experiment on both training and test dataset is Decision tree algorithm, which is able to achieve classification accuracy of 98.95% which is 2.62 % and 4.03% higher than K-NN and SVM respectively



REFERENCES

[1] Dong-Jie Wu, Te-En Wei, Ching-Hao Mao, Kuo-Ping Wu, and Hahn-Ming Lee.,, "Droidmat: Android malware detection through manifest and api calls tracing", Information Security (Asia JCIS), 2012 Seventh Asia Joint Conference on, pp.62–69, 2012.

[2] Ryo Sato, Shigeki Goto, and Daiki Chiba., "Detecting android malware by analyzing manifest files", Proceedings of the Asia-Pacific Advanced Network, pp.23–31, 2013.

[3] Chun-Ying Huang, Chung-Han Hsu, and Yi-Ting Tsai., "Performance evaluation on permission-based detection for android malware", Advances in Intelligent Systems and Applications-Volume 2, pp.111– 120, , 2013.

[4] Borja Sanz, Carlos Laorden, Igor Santos, Xabier Ugarte-Pedrero, Gonzalo Alvarez, and Pablo Garcia Bringas'., "Puma: Permission usage to detect malware in android", International Joint Conference CISIS12-ICEUTE´12-SOCO´ 12 Special Sessions, pp.289–298, 2013.

[5] Wang, W., Wang, X., Feng, D., Liu, J., Han, Z., & Zhang, X., "Exploring Permission-Induced Risk in Android Applications for Malicious Application Detection", IEEE Transactions on Information Forensics and Security, 9(11), pp.1869–1882, 2014.

[6] Xiang Li, Jianyi Liu, Yanyu Huo , Ru Zhang, Yuangang Yao. , "An android malware detection method based on androidmanifest file", Proceedings of CCIS2016, pp.239-243, 2016.

[7] Varma, P. R. K., Raj, K. P., & Raju, K. V. S. , "Android mobile security by detecting and classification of malware based on permissions using machine learning algorithms", 2017 International Conference on I- SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), pp.294- 299, 2017.

[8] Koli, J. D., "RanDroid: Android malware detection using random machine learning classifiers", IEEE international conference on Technologies for Smart-City Energy Security and Power (ICSESP), 2018.

[9] Urcuqui, C., & Navarro, A., "Machine learning classifiers for android malware analysis", Communications and Computing (COLCOM), 2016 IEEE Colombian Conference, pp. 1-6, April 2016.