ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

MACHINE LEARNING APPROACH TO DETECT ANDROID MALWARES

Nikita Rai* and Dr Tripti Arjariya

DOI: 10.5281/zenodo.47585

ABSTRACT

Mobile phone industry is growing at rapid speed .These mobile phones are running on diiferent platform such as JAVA, Android, IOS, Sysmbian and others. Out all these platforms Android cover maximum share amount Smartphone platform. Android platforms supports millions of applications that can be download from various repositories such as google play. These applications are installed and used. The applications present in these repository may be malicious which leady to security problems using these application. In this paper an effective approach has been proposed for detection of the malicious application based on the permission groups. In proposed work, binary classification of applications are carried out into two label i.e. Benign and malicious one. In this developed approach the distinguished features are evaluated and filtered out using features evaluation technique such as Information gain, Gain ratio, Gini Index, Chi-square test. Finally based on the features evaluated the classification is done using supervised machine learning techniques.

Keywords*:* Malicious, Android, Classification, Naïve Bayes

INTRODUCTION

The rapid growth of smartphone has led to a renaissance in mobile application services. Android and I-phone operating system (IOS) are the most common platform for Smartphone. These platforms are having their own market from where required applications can be downloaded. Any application can be downloaded from the App Store (iPhone) or Android Market (Google Android), both of which provide point and click access for hundreds and thousands of users to commercial or free applications.

With an estimated market share of 70% to 80%, Android has become the most popular operating system for Smartphone and tablets. Expecting a shipment of 1 billion Android devices in 2017 and with over 50 billion total app downloads since the first Android phone was released in 2008, cyber criminals naturally expanded their vicious activities towards Google's mobile platform. With the increase demand and vast usage, the security of Android mobile themselves and their application services have become increasingly important issue for mobile owners.

LITERATURE REVIEW

The open natureof the android system has certain benefits and drawbacks. As android source code is available open it becomes very easy for attacker to develop malware which can harm any android device. In this section the work done in the direction of malicious application detection is discussed.For analysing malware different type of techniques has been prposed, but on broader scale these techniques are categorized as:

- Static analysis
- Dynamic analysis
- Hybrid analysis.

Android Malware Forensics: Reconstruction of Malicious Events et.al Juanru Li, DawuGu, YuhaoLuo proposed a systematic procedure for Android malware forensic analysis and malicious events reconstruction.This paper discuss about how to defeat anti forensics code. How to combine existing tools and techniques to help analysis.

Permission-Based Android Malware Detection [10] et.al Zarni Aung, Win it describe the process of extracting features from the Android .apk files. In this paper a new dataset has been created from extracted features of Android applications in order to develop android malware detection framework.

Mobile-Sandbox: Having Deeper Look into Android Application et.al Michael Spreitzen barth proposed a Mobile-Sandbox, system which is designed to automatically analyse Android applications in two novel ways. It combines static and dynamic analysis, i.e., results of static analysis are used to guide dynamic analysis extend coverage of executed code.It uses specie techniques to log calls to native (i.e., \non-Java") APIs.

PRPOSED METHOD

The process for identification of malicious android application consist of step wise approach and steps are divided further into different sub tasks which includes:

a) Data extraction
b) Features extraction
c) Features evaluation
d) Classification techniques

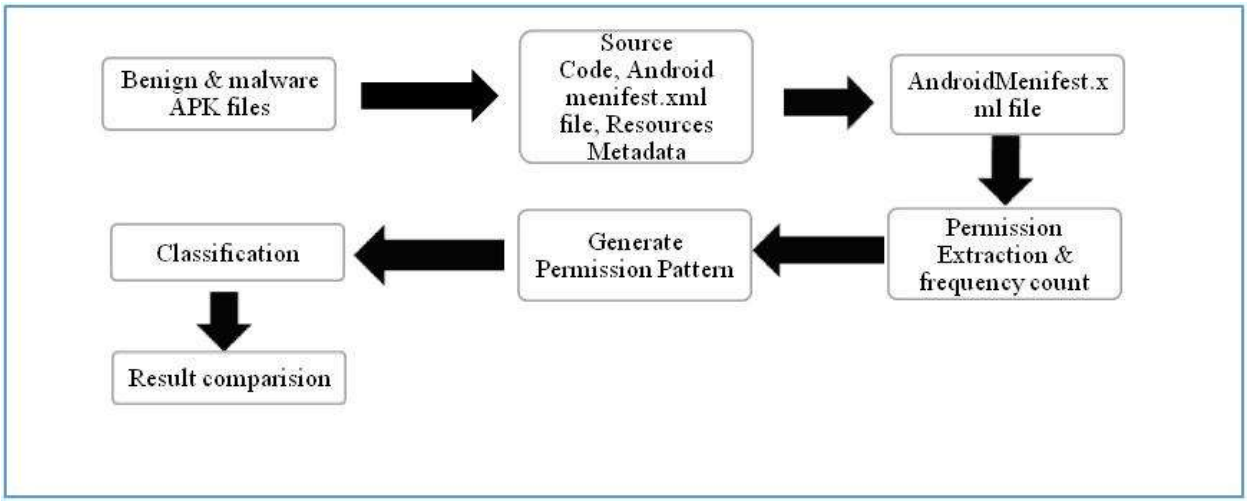


Figure 1: Process flow for malicious application detection

A. Data Extraction:

In this step the different types of APK files are extracted from different repositories. These apk files are special type of compressed files which includes source code, manifest.xml and other resources as required by the application. In this step the apk files are download from Google play[], sharevirus[] for benign and malicious application.

B. Feature Extraction:

It is the most cruituial step for the whole process as classification depends on which features are extracted. In our proposed work permission regading each application are considered as the features as most of the malicious applications uses some commom type of permission pattern which is quite different from the benign applications. The apk file is unzipped and permisiions regarding that aaplication are extracted from the manifest.xml file. For this purpose we develop a xml parser which extract all the permmsion for every application and dataset is built.

C. Feature Evaluation:

The extracted features includes similar and distinguish features for type of application i.e malicious and benign. In order to get better classification result and accuracy there is a need of filter down this features set.

Feature evaluation step find the correlation and calculated amount of information per feature. The features which has no information value are pruned from tha dataset and new refined dataset is made. Moreover this reduced feature set reduce the overburden and provide optimizing result classification.

In the prposed work recurcive feature evaluation technique and cross correlation methods are used for feature evaluation. The feature evaluation methods are implemented with the help of R statistical language.

D. Classification Technique:

Classication techniques are machine learning techniques in which algorithm is first trained with the help of available dataset and then it is tested in terms of correctly classification rate. These classification techniques inludes Naïve bayes, decision tree, Support vector machine and others.

Naïve Bayes classification technique based on the probabilistic approach i.e. Bayes rule. Naïve Bayes is quite hood in malicious filtering domain as there are many equally important features for different class of attributes. It is quite fast learning technique with one pass of counting over the data; testing linear in the number of attributes, and document collection size.

	access_ns	access_ws	r_calender	r_e_storage	vibrate	access_coa	get_acc	r_sync_set	w_sync_se	broadcast_	download_	camera
access_ns	1	0.472826	0.0861514	0.1735539	0.2437692	0.0213524	0.0768498	0.1059544	0.1379468	-0.0376622	0.0382746	0.0299063
access_ws	0.472826	1	0.1082454	0.2146892	0.0508822	0.1140834	0.0706052	0.1437025	0.2146302	-0.0033169	0.0595511	0.0679856
r_calender	0.0861514	0.1082454	1	0.0086113	0.1663218	0.0922225	0.3305091	0.3129817	0.1687392	0.213498	0.2184457	0.2592169
r_e_storage	0.1735539	0.2146892	0.0086113	1	-0.0165477	0.1293909	0.0588699	0.1070414	0.0589949	-0.0141169	0.0874454	0.1175277
vibrate	0.2437692	0.0508822	0.1663218	-0.0165477	1	-0.072179	0.1667938	0.2160233	0.3111176	0.0605581	0.0863224	0.1150848
access_coa	0.0213524	0.1140834	0.0922225	0.1293909	-0.072179	1	0.0576557	0.0921546	0.1808931	-0.0311265	0.0640184	0.0558052
get_acc	0.0768498	0.0706052	0.3305091	0.0588699	0.1667938	0.0576557	1	0.482278	0.2627695	0.1267886	0.1879068	0.3759471
r_sync_set	0.1059544	0.1437025	0.3129817	0.1070414	0.2160233	0.0921546	0.482278	1	0.6079231	0.1661178	0.3612365	0.2950657
w_sync_se	0.1379468	0.2146302	0.1687392	0.0589949	0.3111176	0.1808931	0.2627695	0.6079231	1	0.1158124	0.1328212	0.1740229
broadcast_	-0.0376622	-0.0033169	0.213498	-0.0141169	0.0605581	-0.0311265	0.1267886	0.1661178	0.1158124	1	0.1645375	0.1389617
download_	0.0382746	0.0595511	0.2184457	0.0874454	0.0863224	0.0640184	0.1879068	0.3612365	0.1328212	0.1645375	1	0.243775
camera	0.0299063	0.0679856	0.2592169	0.1175277	0.1150848	0.0558052	0.3759471	0.2950657	0.1740229	0.1389617	0.243775	1

Figure 2: Feature corelation matrix

Naive bayes classification is based on following values:

i. ***Prior probability:*** The probability that an event will reflect established beliefs about the event before the arrival of new evidence or information. Prior probabilities are the original probabilities of an outcome, which will be updated with new information to create posterior probabilities.
ii. ***Posterior probability:*** The revised probability of an event occurring after taking into consideration new information. Posterior probability is normally calculated by updating the prior probability by using Bayes' theorem. In statistical terms, the posterior probability is the probability of event 'A' occurring given that event B has occurred.
iii. ***Bayesian probability:*** This is based on prior and posterior probability of the events.
iv. ***Independent events probability:*** The probability of two independent events is defined by the product of their individual probability.

$$P(A \wedge B) = P(A) \times P(B) \cdots\cdots\cdots\cdots (1)$$

v. ***Conditional Probability:*** The probability value which depends on the sequence of two events.

$$P(A|B) = \frac{P(A \wedge B)}{P(B)} \cdots\cdots\cdots\cdots (2)$$

And

$$P(B|A) = \frac{P(A \wedge B)}{P(A)} \cdots\cdots\cdots\cdots (3)$$

From equation 2 and 3, P (A^B) = P (B|A).P (A).
In R language the Naïve bayes technique is implemented.

EXPERIMENTAL SETUP

For the classification purpose R studio[13] version 0.99.484 and Java runtime version1.8.0_31-b13 with system configuration as Intel I-5 processor 3rd generation with 8 GB of RAM memory is used. R studio provide a developing

environment for R language. R language is a collection of machine learning algorithms for data mining tasks. The classification algorithms can either be applied using R libraray, packages and interfaces. R languages contains packages for data pre-processing, classification, regression, clustering, association rules, and visualization. R is open source software issued under the GNU General Public License.

In experimental setup, dataset of malicious and benign android aaplications randomly divided into almost 10 sets for cross validation i.e. Dataset ={D_1, D_2, D_3, D_4, D_5, D_6, D_7, D_8, D_9, D_{10}}. All the sets are mutually exclusive. Out of these 10 sets 9 sets at each iteration are used for training purpose i.e. $E_{training}$ and remaining set is used for testing i.e. $E_{testing}$. This process is repeated 10 times. At each iteration i.e.:

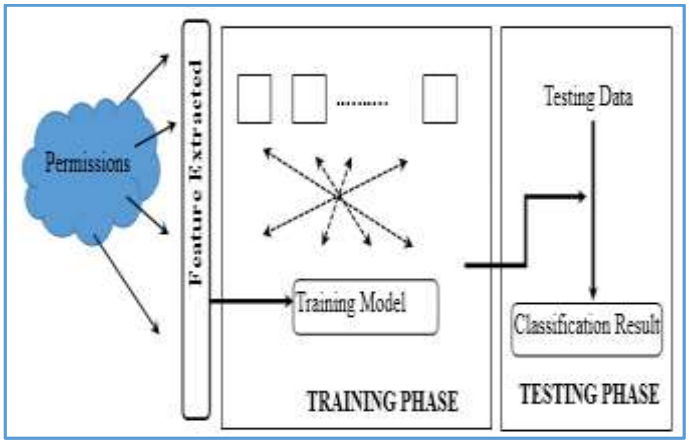


Figure 3: Cross validation and Classification

First iteration: {D_2, D_3, D_4, D_5, D_6, D_7, D_8, D_9, D_{10}} for training and D_1 for the testing purpose.

Second iteration: {D_1, D_3, D_4, D_4, D_5, D_6, D_7, D_8, D_9,D_{10}} for training and D_2 for the testing purpose. Usually 10-fold cross validation results to relatively low bias and variance.

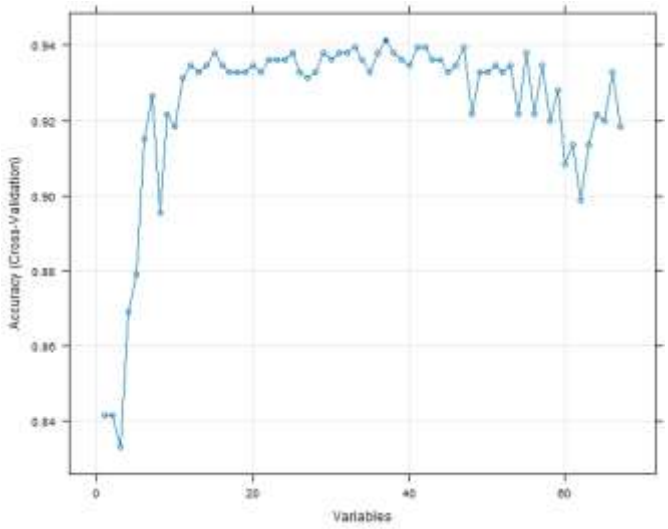


Figure 4: Classification rate as per features.

A. Machine Learning Techniques:

C5.0: It is a classification techniques which information gain parameter for decision. The decision tree generated for the dataset has several levels based on different features. The decision values at each level is described as in figure.

This classification technique is a variation of decision tree is best-known learning algorithm. C50 is re-implemented in R.This method is generally used for binary classification. Unlike nominal attributes, every attribute has many splitting points but uses gain ratio technique for feature selection.

***Table** Error! No text of specified style in document..1: 2X2 Confusion matrix for C50 classification technique*

Confusion Matrix		Predicted Values	
		Benign	**Malware**
Actual Values	**Benign**	29	4
	Malware	5	166

The accuracy rate for C50 is 91.47%

```
GET_ACCOUNTS.numeric> 0::... ACCESS_NETWORK_STATE.numeric<= 0: Malware (6)
:   ACCESS_NETWORK_STATE.numeric> 0: Benign (29/3)
GET_ACCOUNTS.numeric<= 0:
:...SEND_SMS.numeric> 0: Malware (185/1)
SEND_SMS.numeric<= 0:
:...INSTALL_PACKAGES.numeric> 0: Malware (60/1)
INSTALL_PACKAGES.numeric<= 0:
:...CAMERA.numeric> 0: Benign (7)
CAMERA.numeric<= 0:
:...READ_SMS.numeric> 0: Malware (28/2)
READ_SMS.numeric<= 0:
:...CHANGE_WIFI_STATE.numeric> 0: Malware (37/3)
CHANGE_WIFI_STATE.numeric<= 0:
:...ACCESS_COARSE_LOCATION.numeric> 0: Benign (9/1)
ACCESS_COARSE_LOCATION.numeric<= 0:
:...RECEIVE_BOOT_COMPLETED.numeric> 0: Benign (5)
RECEIVE_BOOT_COMPLETED.numeric<= 0:
:...UPDATE_APP_OPS_STATS.numeric> 0: Benign (2)
UPDATE_APP_OPS_STATS.numeric<= 0:
:...ACCESS_WIFI_STATE.numeric> 0: Malware (13/4)
ACCESS_WIFI_STATE.numeric<= 0:
:...ACCESS_NETWORK_STATE.numeric<= 0: Malware (12/3)
ACCESS_NETWORK_STATE.numeric> 0: Benign (15/1)
```

B. Random Forest:

It is improved technique of decision tree. Its is an ensemble model combinethe results from different models.The result from an ensemble is better than the result of individual model.In this work different classification techniques has been applied on the basis of results and accuracy the classification result of Random forest method is fairly good with classification accuracy as 98.31%. The absolute mean error is 0.0635.

RESULT ANALYSIS

It is the measure to how accurately the training model classified the test data set. This correct can be measured four value i.e. TP, TN, FP, FN directly or indirectly. Random forest gives the best classification results as comparision to C50 and E1071(Naïve bayes).

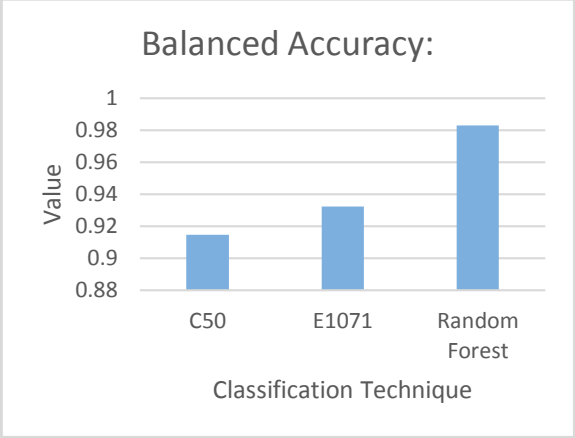
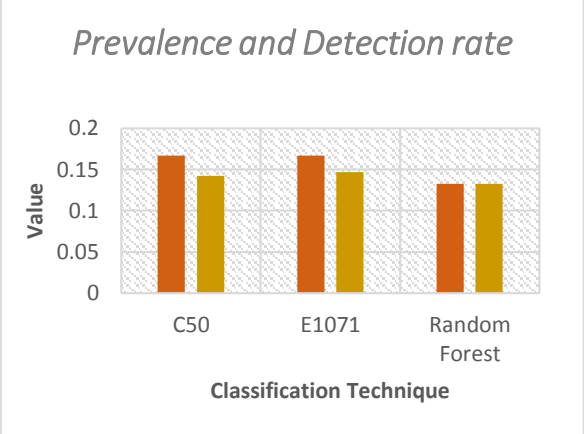


Figure 5: Classification accuracy for different classification techniques



Figutre6: Comparison based on Prevalence and Detection rate

Finally developed approach is analysed through calculating the prevalence and detection rate. Prevelance is defined as a fraction, or a percentage or as the number of malicious application detected per 1000 dataset. The detection rate is defined as the number of malicious application detected by the system (True Positive) divided by the total number of intrusion instances present in the dataset.

CONCLUSION AND FUTURE WORK

Malicious application are one the main barrier of today's mobile security infrastructure. Malicious application is a collective term coined for all those application which are either themselves or support other aaplicationfor various attacks. Some of the common android mobile based attacks are data leak, password theaft, malware and others.

The work presented includes the comparison of different feature evaluation and classification techniques under android application scenario. The comparison of features evaluation is done in order to identify the minimum and optimize set of feature vector.

In classification, random forest updateable comes out with best result for multi class classification as compared to Decision tree (C50) and Naïve bayes (E1071). In this work different classification techniques has been applied on the basis of results and accuracy the classification result of Random forest method is fairly good with classification accuracy as 98.31%. The absolute mean error is 0.0635.

REFERENCES

[1]. Sonal Nerurkar, "Teens drive Indian smartphone sales, study finds" online[Available], http://timesofindia.indiatimes.com/business/indiabusiness/Teens-drive-Indian-smartphone-sales-studyfinds/articleshow/22406572.cms [Accessed : 23 Nov 2013]
[2]. Cesare, x. yang and silvio, "Classification of malware using structured control flow," Australian Computer Society, vol. 107, pp. 61-70, 2010.
[3]. Spreitzenbarth, Michael and F. Felix, "Mobile-sandbox: having a deeper look into android applications," Annual ACM Symposium on Applied Computing, pp. 1808-1815, 2013.
[4]. "Apktool download," [Online]. Available: code.google.com/p/android-apktool/. [Accessed 11 03 2015].
[5]. Allix, kevin and q. jerome, "A Forensic Analysis of Android Malware--How is Malware Written and," compsac, pp. 384-393, 2014.
[6]. "Android malware hijacks power button," [Online]. Available: http://www.theregister.co.uk/. [Accessed 12 5 2015].
[7]. Machiry, Aravind and T. Rohan, "Dynodroid: An input generation system for android apps," in Foundations of Software Engineering, 2013.
[8]. Aung, Zarni and Z. Win, "Permission-based Android malware detection," International Journal of Scientific and Technology Research, pp. 228-234, 2013.

[9]. Barrera, David and G. H., "A methodology for empirical analysis of permission-based security models and its," ACM conference on Computer and communications security, pp. 73-84, 2010.
[10]. Min, X. Luo and H. C. Qing, "Runtime-based behavior dynamic analysis system for android malware detection," In Advanced Materials Research, vol. 756, pp. 2220-2225, 2013.
[11]. M. Spreitzenbarth et Al., "Mobile-Sandbox: Having Deeper Look into Android App.", SAC'13 March-2013, Coimbra, Portugal, pp. 18-22.
[12]. Juanru Li, DawuGu and YuhaoLuo, "Android Malware Forensics Reconstruction of Malicious Events", In 32th Int. Conf. on Distributed Computing Systems Workshops 2012.
[13]. Lei Cen et Al., "A Probabilistic Discriminative Model for Android Malware Detection with Decompiled Source Code", Pub. In IEEE Trans. on dependable and secure computing, Vol.12, Issue. 04, 2015, pp. 400-412.
[14]. Zhang, yuan and y. min, "Permission Use Analysis for Vetting Undesirable Behaviors in Android Apps," Information , pp. 1828-1842, 2014.
[15]. Zarni Aung and Win Zaw, "A Probabilistic Discriminative Model for Android Malware Detection with Decompiled Source Code", Int. J. of Scientific & Technology Research, Vol. 2, Issue 3, March 2013, pp. 228-234.
[16]. Dai-Fei Guo et Al., "Behavior classification based self-learning mobile malware detection", Pub in J. of Computers, Vol. 9, Issue 4, 2014, pp. 851-858.