MALWARE DETECTION IN ANDROID APP STORE USING ROBUST ATTRIBUTE GENERATION

Shivani Y M[1], **Dr. Ch Ramesh**[2]
Computer Network and Information Security, Jawaharlal Nehru Technological University
[1] Department of Information Technology, GNITS, Telangana, Hyderabad, yalalashivani@gmail.com
[2] Assistant Professor, Department of Information Technology, GNITS, Hyderabad
chramesh23@gmail.com

ABSTRACT: Now a days usage of android devices is becoming quite more due to day-to day increasing the number of active users. With the first-class platform for create gaming and applications android as gained admiration among all other smartphone. It will also allow users to sell and distribute apps instantly and also offers ample free third-party download and install application from the Google play store and third-party play store. Today more the 350,000 sample of new malware are found per every day and 5.2% increase compared to last year.

Mobile devices can control and track the Internet of Things (IOT) services and make them prone to attacks by various malicious application. Conventional methods fail to detect malicious application because of the growth in numbers, variants, and advancement of the malware. Moreover, android allows to installation and downloading apps from unverified sources. For detecting malware application current solution are not appropriate with the rise of the malware and there are limited by low detection accuracy, advanced implementation and heigh computational cost and power. Therefore, android malware become a real-world challenge. To solve this problem an automatic intelligence technique is required for detecting malware apps with the use of real-word datasets which analyses the source code. To show the uniformities in apps which hold malware content we proposed API calls and use Chi- square for SelectKbest best feature selection and examined combination with six supervised machine-learning algorithms (Random Forest, Decision Tree, K-Neighbours, AdaBoost, SVM Linear, and Naïve Bayes). The detection accuracy of these six algorithms is analyse to identify the most effective classifier for detecting malware.

key words: malware, android devices, machine learning, android features, Feature extraction, features selections, Random Forest, Decision Tree, K-Neighbours, AdaBoost, SVM Linear, Naïve Bayes and chi-square

I. Introduction:

Android has become the most common operation system for tablets, MacBook and smartphone with an estimated market share of 70% to 80%. Android has overtaken many other mobile operating systems to become one of the most popular mobile platforms in the world. Recognition of smartphone and other kind of mobile devices has drop down significantly. Android has gained huge admiration among all other smartphone it is the first-class platform for creating gaming and application which allows to distribute and also offer free third party's apps to install and download from Google play store for selling and distributing android apps.

The Internet of Things (IoT) is an attractive system that connects several logical objects and physical devices with networks to enlarge their communication capabilities. In previous years, the IoT has gained popularity owing to technological advancements in areas like artificial intelligence, cloud computing, machine learning, deep learning, application system, and smart home devices. Now a days mobile users are interested in using online transaction for example using smartphone for paying bills, online shopping, trading, booking tickets etc. such portable device is targeted by hacking and become vulnerable to attack more and more.

Malware refer to malicious code which harms user integrity, availability, and confidentiality. Malicious activity is hidden in the background and appear as a clean application. Some examples of the android malware are stealing user personal information (e.g., bank account number, bank credentials, contact number, passwords), tracking user location, send premium SMS messages that cost more than the standard ones, streaming videos from users' cameras, send SMS or mails which contain malware links, and encrypting personal information such as video, images, SMS and contact.

The shared report by IDC for smartphone increases 25.5% global share market in the 1rd quarter of the year 2021[1], the total market share of Android was 72.83% [2] and more than 3.8 billion active users around the world in 2021.

Android is one of the most popularly used mobile device in the world, because of its framework, compatibility, technological impact, open-source, higher success ratio, inter app

integration, high level multitasking, user friendly development environment, etc. On the other hand, it also has risk in installing and downloading applications from unauthorize web site or third-party. Because of the open -source applications android is getting more prone to attack. It has become target to malware, even though it provides security mechanisms.

To prevent malware attacks, developers and researchers are working in different security solution by applying data science, artificial intelligence, machine learning and deep learning. We can analysis android malware with two techniques static analysis and dynamic analysis approaches. In static analysis we get information about software which are going to analyses without executing it but ware as in dynamic analysis is doing during time of execution.

II. RELATED WORK

SNO	Description	Result
	The authors [3] introduce SIGPID and use 3 level pruning to identify signification permission	SIGPID can detect 93.63%of malware in the dataset and 91.4% unknown malware
2	The authors [4] proposed network traffic and consider seven feature extractions	Drebin feature extraction get heigh accuracy rate we can to other six feature extraction
3	The authors [5] used TF-IDF to calculate PV and SVOA	The proposed approached get more accuracy against other state-of-art.
4	The authors [6] use Chi-square test for permission feature in machine learning classifiers.	Rate of missing malware detection is 10.33% and overall detecting accuracy is 88.9%
5	The authors [7] proposed to combine permission and API to use machine learning classifiers to detect malware.	Detected 1200 malware apps and 1200 benign apps
6	The authors [8] proposed extracted feature vector from Android Manifest file and combines PI and CI	Results shows better against other traditional permission.
7	The authors [9] use state-of-the-art work which different features to charactering behaviour to detects malware applications	Features are available for State-of-the-art, deep learning technique are used to detect malware.
8	The authors [10] use lightweight for metadata Mmda to statically analyse.	Random forest gives 94%of accuracy.
9	The authors [11] evaluated effective android internet for feature for identifying malware apps	Detection rate for android internet is 91% against android permission with 83%
10	The author [12] use traditional way of Principal Component Analysis and feature are extracted from DEX	Detection accuracy of state-of-the-art is 88.24%

Table1: Related Work

III. PROPOSED SYSTEM

In this project, we are focusing on analysing the malware using supervise machine learning algorithms
and comparing accuracy rate of each classifier. Fig 1 show the android malware detection modules.
1. Dataset
2.Feature extraction
3.Feature vector
4.Feature selection
5. Training/Testing
6.Algorithm classification
7. Malware detection
8.Algoritham comparison

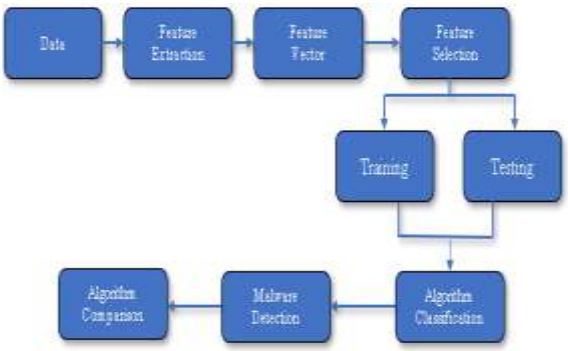


Fig 1: Models for Android Malware Detection

Dataset:

In the first module we collected real-world malware from android security and AndroZoo which are in the form of binary (0 and 1).in the dataset 0 means clean and 1 means malware feature. Table 1 gives a brief about the dataset used for experiments.

SNO	Application type	Total number of application
1	Malware	199
2	Benign	250

Table 2: Dataset Description

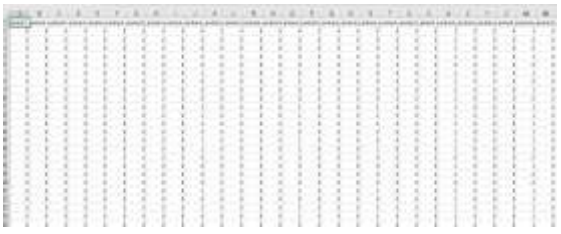


Fig 2: Dataset

Feature extraction:

In this module we convert android package (.APK) to AndroidManifest.xml and then to csv file by using apktool[13]. AndroidManifest.xml content features which are used in static analysis

Clean	
Permission	Total no
android.permission.INTERNET	104
android.permission.WRITE_EXTERNAL_STORAGE	76
android.permission.ACCESS_NETWORK_STATE	62
android.permission.WAKE_LOCK	36
android.permission.RECEIVE_BOOT_COMPLETED	30
android.permission.ACCESS_WIFI_STATE	29
android.permission.READ_PHONE_STATE	24
android.permission.VIBRATE	21
android.permission.ACCESS_FINE_LOCATION	18
android.permission.READ_EXTERNAL_STORAGE	15

Table 3 shows the permissions supplied in the Android manifest for each clean app

Malware	
Permission	Total no
android.permission.INTERNET	195
android.permission.READ_PHONE_STATE	190
android.permission.ACCESS_NETWORK_STATE	167
android.permission.WRITE_EXTERNAL_STORAGE	136
android.permission.ACCESS_WIFI_STATE	135
android.permission.READ_SMS	124
android.permission.WRITE_SMS	104
android.permission.RECEIVE_BOOT_COMPLETED	102
android.permission.ACCESS_COARSE_LOCATION	80
android.permission.READ_EXTERNAL_STORAGE	30

Table 4 shows the permissions supplied in the Android manifest for each malware app.

Feature vector:

In feature extraction module we consider V as vector which contain set of android permission $vi = v1, v2 \dots vj$

$$vi = \begin{cases} 1, & \text{when permission exist.} \\ 0, & \text{otherwise.} \end{cases}$$

Feature selection:

In this feature selectin module, we will compare feature selection algorithms with different method for reduce the feature-vector.

Training/Testing:

In testing and training module first, we train the model with feature train the 300 datasets

with features to detected malware and then test the 500 datasets for more accuracy.

Algorithm classification and Malware detection:

In this model we study about detailed implementations of supervised machine learning algorithms such as

- Random Forest
- Decision Tree
- K-Neighbours
- AdaBoost,
- SVM Linear, and
- Naïve Bayes.

Using this classification, we detected malware.

Random Forest classifiers:

The random forest is a classification technique that uses multiple decision trees to classify data. When developing each individual tree, it employs bagging and feature randomization in order to produce an uncorrelated forest of trees whose committee prediction is more accurate than that of any one tree.

Decision Tree:

The classification model is built using the decision tree method in the form of a tree structure. It employs if-then principles, which are both exhaustive and mutually exclusive when it comes to categorization. The process continues with the data being broken down into smaller structures and finally being linked to an incremental decision tree. The finished product resembles a tree with nodes and leaves. The rules are learnt one by one, one by one, utilising the training data. The tuples that cover the rules are deleted each time a rule is learnt. On the training set, the procedure continues until the termination point is reached.

K-Nearest Neighbour:

The K-Nearest Neighbour (KNN) method is a supervised learning technique that may be used to solve regression and classification issues. It's most commonly used in machine learning for categorization issues. KNN is based on the premise that every data point that is close to another belongs to the same class. In other words, it uses similarity to classify a new data point. It's an n-dimensional space-based lazy learning method that saves all instances corresponding to training data. It's a lazy learning method since it focuses on keeping instances of training data rather than building a broad internal model.

AdaBoost Classifier:

Adaptive Boosting, short for Adaptive Boosting, is a Boosting approach used in Machine Learning as an Ensemble Method. The weights are reallocated to each instance, with greater weights to erroneously categorised occurrences. This is termed Adaptive Boosting. In supervised learning, boosting is used to decrease bias and variation. It is based on the idea of successive growth of learners. Each succeeding student, with the exception of the first, is produced from previously grown learners. In other words, weak students are transformed into strong students. Although the Adaboost algorithm works on the same concept as boosting, there is a little variation in how it operates.

Support Vector Machine:

SVM (Support Vector Machine) is a supervised machine learning method that may be used to solve classification and regression problems. It is, however, mostly employed to solve categorization issues. The value of each feature is the value of a specific coordinate in this technique, which plots each data item as a point in n-dimensional space (where n is the number of characteristics you have). Following that, we do classification by locating the hyper-plane that best distinguishes the two classes. Individual observation co-ordinates are what Support Vectors are. The Support Vector Machine is a frontier that separates the two classes (hyper-plane and line) the best.

Algorithm:

Step1: Initialization
F: a set of features
A: a set of Android applications
C: a set of classifiers ∈ {Naïve Bayes, Random Forest, k-NN, SVM, decision tree, AdaBoos.}
TSD: a set of testing datasets
TRD: a set of training datasets
B*: Boolean score // 0 or 1
Labels: L = {Malicious, Clean}

Step 2: Extract features F
For each feature Fi in F
Count_freq Fi ();
Freq [Fi] + = Freq [Fi]
PF ← Pf/|A|
End for
For each PF :

Step 3: Select top selected features F using Chi-Square

$$\sum_{i=1}^{k} ni\ Yi - Y\ 2\ /\ (K - 1)$$

For each classifier Ci in C*:

Step 4: Test classifier (Ci , tsdi)
End for
For each classifier Ci in C*:

Step 5: Train classifier (Ci , trdi)
ri = classify (Ci , VDi)

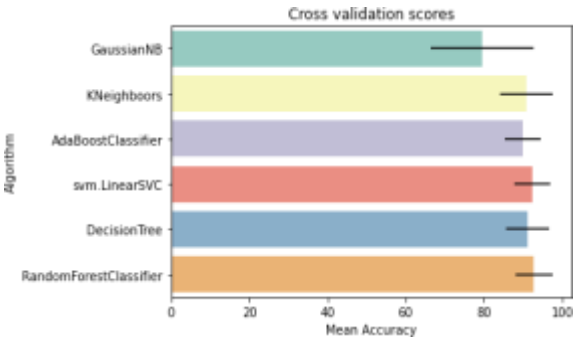
Step 6: Application label. Add (labeli)
End for

Algorithm comparison:

In this module we compare above mention six classification and analysis most effective classifier for given dataset.

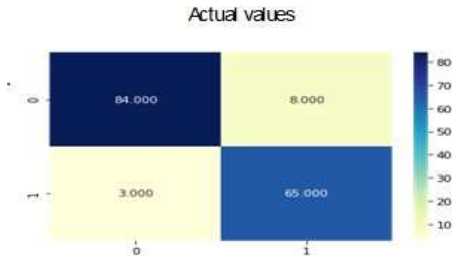
IV. RESULTS:

In this section, we discuss our results and the main aim of this project is to compare the results which are obtained from six algorithms used in project. Fig 2 show the brief discussion about algorithm and detection accuracy of Naïve Bayes 88%, 91% for K-Neighbours, 93% for AdaBoost, 93% for SVM Linear,93% for Decision Tree and 95% for Random Forest. Among all classification we get best score for random forest classifier.



FIGER 2: Algorithm comparation

We also calculate confusion matrix which compares the active target value with those predicted values by the machine learning model confusion matrix which of four values such as True positive, true negative, false positive and false negative Fig 3 show the confusion matrix



FIGER 3: Confusion matrix

Chi2 Testing:

There are two kinds of chi-square testing. For distinct purposes, both employ the chi-square statistic and distribution. The chi-square goodness of fit test assesses whether or not sample data is representative of the population. In a contingency table, a chi-square test for independence examines two variables to discover if they are linked. In a broader sense, it examines whether categorical variable distributions differ from one another.

$$x^2 = \sum \frac{(O-E)^2}{E}$$

Where x^2 denotes freedom of degree.
O denotes the observed value.
E denotes the expected value.

Chi-Square uses:

1. Estimation of a population standard deviation of a normal distribution from a sample standard deviation using confidence intervals.
2. Two categorization criteria for qualitative variables are independent.
3. Categorical variables and their relationships.
4. When the underlying distribution is normal, sample variance analysis is used.

V. CONCLUSION:

In this project we proposed permissions, API calls, and a malware detection model that is capable of quick, generalised, accurate, and efficient detection of Android malware in Android applications in android platforms. To achieve this goal, we've created a set of modules that address the problems of detecting Android malware. The datasets we work with are highly class balanced, machine learning models should be able to train well on them.

We analysed the usefulness of many kinds of data, including as permissions, APIs, Intents, and App Components, in detecting Android malware. SelectKBest feature selection approaches were used to identify the relevant characteristics that are the most informative and crucial for malware detection. Detection accuracy of Naïve Bayes 90%, 94% for K-Neighbours, 93% for AdaBoost, 95% for SVM Linear, 94% for Decision Tree and 96% for Random Forest. We analysed to identify the most effective classifier for detecting malware is random forest.

VI. Future Work:

We have only looked at characteristics derived via static analysis in this paper. Dynamic analysis may also be used to extract

more features, resulting in a more useful feature set. Furthermore, in the future, we will be able to test this malware detection in a variety of ways based on our technological advancements.

We intend to decrease false positives and negatives in the future by analysing samples that were incorrectly categorised and determining the causes for the misclassification.

VII. REFERENCES

[1] IDC smartphone market share https://www.idc.com/getdoc.jsp

[2] Android share market: https://gs.statcounter.com/os-market-share/mobile/worldwide

[3] Lichao Sun; Zhiqiang Li; Qiben Yan; Witawas Srisa-an; Yu Pan: "SigPID: significant permission identification for android malware detection",2016. https://ieeexplore.ieee.org/document/7888730/authors

[4] Aqil Zulkifli;Isredza Rahmi A. Hamid; Wahidah Md Shah; Zubaile Abdullah: "Android Malware Detection Based on Network Traffic Using Decision Tree Algorithm",2018.

[5] Hongli Yuan;Yongchuan Tang; Wenjuan Sun; Li Liu: "A detection method for android application security based on TF-IDF and machine learning",2020.

[6] R Zhang; J Yang; "Android malware detection based on permission correlation",2014.

[7] Naser Peiravian; Xingquan Zhu: "Android malware detection using machine learning using permission and API call",2013.

[8] Xiang Li; Jianyi Liu; Yanyu Huo; Ru Zhang; Yuangang Yao : "An Android malware detection method based on Android Manifest file",2016.

[9] Wei Wang; Meichen Zhao; Zhenzhen Gao; Guangquan Xu; Hequn Xian; Yuanyuan Li; Xiangliang Zhang: "Constructing Features for Detecting Android Malicious Applications: Issues, Taxonomy and Directions",2019.

[10] Kun Wang; Tao Song; Alei Liang:"Metadata based malware detection on android. in computational intelligence and security",2016.

[11] AliFeizollah; Nor BadrulAnuar; Rosli Salleh; Guillermo Suarez-Tangil; Seteven Furnell:"AndroDialysis: Analysis of Android Intent Effectiveness in Malware Detection",2017.

Luiza Sayfullina;Emil Eirola;Dmitry Komashinsky: "Android Malware Detection: Building Useful Representations. In Machine Learning and Applications",2016

[13] Virustotal, Retrieved from https://www.virustotal.com

[14] Android OS market share of smartphone sales to end users from 2009 to 2020; https://www.statista.com/statistics/216420/global-market-share-forecast-of-smartphone-operating-systems/.

[15] Zhang R, Yang J. Android malware detection based on permission correlation. Journal of Computer Applications. 2014;

[16] Yang G, Huang J, Gu G. Automated Generation of Event-Oriented Exploits in Android Hybrid Apps. In: Network and Distributed System Security Symposium; 2018.

[17] Malware statistics and trends report 2021 https://www.av-test.org/en/statistics/malware/

[18] cyber security statement 2021: https://purplesec.us/resources/cyber-security-statistics/

[19] Scikit learning machine learning-python: https://scikit-learn.org/stable/

[20] Machine learning master: https://machinelearningmastery.com/types-of-classification-in-machine-learning/