International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.089|

||Volume 9, Issue 4, April 2020||

An Effective Machine Learning Aided Optimized Features for Android OS to Detect Malware

B. Navya Surya Bhavani[1], Dr. B . Kezia Rani[2]

M.C.A Student, Adikavi Nannaya University, University College of Engineering, Rajamahendravaram, Andhra Pradesh, India

Assistant Professor, Adikavi Nannaya University, University College of Engineering, Rajamahendravaram, Andhra Pradesh, India

ABSTRACT: As Android operating system usage is quickly increasing by people their is high development in software applications which are available through open source framework. The focus of malware attackers turned to the android OS. The paper proposes an adequate machine learning algorithms for Android Malware Recognition utilizing Extra Tree Classifier for prejudicial features selection .The new model is build to detect malware with selected features. The machine learning classifiers such as k-Nearest Neighbours, Random Forest and decision Tree classifiers are used. The preliminary outcomes validate that Extra Tree Classifier gives most upgraded Feature subset helping in reduction of Feature selection. The change in accuracy with features selected and feature subset obtained through Extra tree classifier is checked.

KEYWORDS: Android malware, features selection, machine learning algorithms, Extra Tree classifier.

I. INTRODUCTION

Malware has demonstrated to be a genuine drawback for the portable stage because of pernicious applications are frequently circulated to those gadgets through partner application advertise. Clients will download/transfer the APK records from outsider servers and can introduce into their cell phones. A large portion of the applications from confided in sources zone unit not malware, yet the outsider server giving malware in changed APK. So before the applications region unit being placed in the great phones they'll be identified whether they region unit is existed with malware. The enormous development of cell phone utilization makes it an objective for noxious assailants to engender malware and perform different malignant assaults. Malware as malignant application that can be placed in cell phones, with the expectation of breaking gadget security arrangement concerning privacy, honesty and accessibility of data. To moderate these security dangers, different versatile explicit Detection Systems have been arranged. There are two main approaches when analyzing applications and determining their malicious, namely static analysis and dynamic analysis. While the previous is centered around separating trademark information from the document containing the application, dynamic investigation focuses on its execution, checking the activities performed. Static examination, then again, permits to dissect applications all the more rapidly, yet is intensely influenced by the utilization of jumbling methods .Dynamic investigation is additionally helpless of being futile when the application can recognize that it is being analyzed.

The principle investigation systems for comprehension malware conduct are:
Static Analysis: These sorts of investigations study a program from its source code or dismantling code. In Android case, the beginning stage is the dex code. The regular static systems extricate the program control stream diagram, the opcode and factors varieties, among others, and concentrate the conduct of different branches so as to recognize suspicious code.

Dynamic Analysis: These methodologies depend on examining the malware as per its execution conduct. The regular arrangement depends on an emulator or virtual machine running the application.Different data is removed from this process, specially centered around follows, organize bundles, memory change and register adjustment, among others.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.089|

||Volume 9, Issue 4, April 2020||

Half and half Analysis: These techniques consolidate static and dynamic examination so as to supplement one another.

II. EXISTING SYSTEM

VirusTotal is a website created by the Spanish security company HispasecSistemas.Propelled in June 2004, it was procured by Google Inc. in September 2012. The organization's proprietorship exchanged in January 2018 to Chronicle, an auxiliary of Alphabet Inc. VirusTotal totals numerous antivirus items and online sweep motors to check for infections that the client's own antivirus may have missed, or to confirm against any bogus positives. Records up to 550 MB can be transferred to the site, or sent through email (max. 32MB). Hostile to infection programming sellers can get duplicates of documents that were hailed by different outputs yet passed by their own motor, to help improve their product and, by expansion, VirusTotal's own ability. Clients can likewise check speculate URLs and search through the VirusTotal dataset. VirusTotal for dynamic investigation of malware utilizes Cuckoo sandbox. VirusTotal was chosen by PC World as extraordinary compared to other 100 results of 2007.our online infection scanner doesn't acknowledge records greater than 128 MB. One can't put forward more than the each archive in the turn. Archives are acknowledged to the line, however our free online infection scanner won't disclose to you which individual record was hailed as positive you will just know there was such a record. On the off chance that you need to discover which document was hailed, you should present the records exclusively.

III. LITERATURE SURVEY

Android clients are powerless to pernicious applications that can hack into their own information because of the absence of careful monitoring of their in-gadget security. There have been numerous chips away at formulating malware identification strategies. Some of the essential strategies that are typically followed for conducting malware investigation on cell phones are Static Analysis and Dynamic Analysis.

[1].Yang, Chao, et al. utilized Dagger a lightweight system to powerfully break down the touchy conduct in Android applications. Their framework depends on watching the application's runtime connection with the phone stage utilizing the three kinds of low-level execution data framework calls, Android Binder exchanges, and application process subtleties. The collected data is along these lines utilized for the construction of an information provenance chart which is utilized for recognizing the behavior of uses by coordinating their provenance graph with the conduct diagram designs acquired from the working logic of the Android framework.

2].Schmidt et al. proposed the investigation of static function calls from doubles by applying a bunching calculation to the collected follows. This system was utilized to distinguish malwares in Symbian OS relying upon the cell phone's requirements, such as gadget efficiency, speed and restricted asset usage. Anomaly identification was likewise proposed to recognize malware in Symbian gadgets.

[3].Li et al. proposed DroidADDMiner, an efficient and precise framework to recognize, order and describe the Android malware. DroidADDMiner was an AI based system which removed highlights dependent on the information dependency between delicate APIs. It extricated the API information dependence paths implanted in an application to develop the element vectors used in AI. DroidADDMiner decreased the runtime as well as portrayed the malware's conduct automatically. For malware recognition, DroidADDMiner accomplished detection with high rate. Some creators give an outline of the Android security model.

[4].One of the most significant safety efforts of Android gadget is the authorization based security model. Each application specifies which assets of the gadget needs to be utilized, and the client allows or denies its installation based on the consents required. Breaking down and enforcing the Android's consent security model has been proposed by different creators.

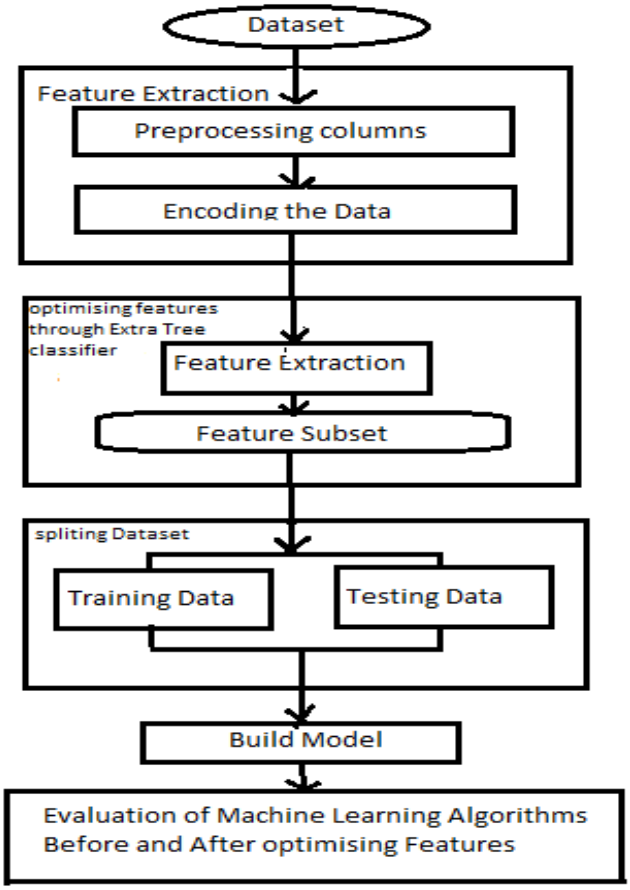
[5]. Regardless of whether a client can be cautioned about the risk of having acknowledged suspicious consents, the spreading of genuine malware has shown that clients straight forwardly trust any application ask for and introduce them on their phones.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.089|

||Volume 9, Issue 4, April 2020||

IV. PROPOSED METHODOLOGY

In proposed framework, android malware dataset is stacked and improve the highlights from original feature set to the feature subset. The following python libraries such as pandas, numpy, matplotlib, scipy and scikit-learn are used. The proposed scheme implements the detection of Android malware with optimised features. Arrangement of Android Apps or APKs: Malware are figured out to separate highlights, for example, authorizations and check of App Components, for example, Activity, Services, Content Providers, and so on. These highlights are utilized as highlight vector with class marks as Malware spoke to by 1 in CSV group. The dataset is saved as 10 .CSV format files, the total of 1000 rows and 85 columns are used which are optimized original features into features subset .further splitting for train ,testing .



Dataset: Dataset involves collection of files or documents which is in form of database tables with rows and columns. Within the dataset rows represents records, and columns represents the variables.
The dataset is collected of different android malwares are saved in .CSV format which are given an limit to each and gathered into single data frame of 1000 rows and 85 columns.

Feature Extraction: The loaded dataset undergoes preprocessing and encoding.The steps involved to feature extraction from each malware application.1.Downloaded and collected different malware applications from application market.2. Decompress applications to extract the contents. 3.Extract permission request of features from each application.4. The dataset is build in .CSV file format.

Preprocessing the columns: In the preprocessing the repeated values in the data set are processed unique values and the unwanted data is removed along with dropping useless columns from dataset.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.089|

||Volume 9, Issue 4, April 2020||

Encoding the features: As the python won't accepts catogerial data it only accepts numerical data. The dataset with catogerial data is transformed into numerical data.

Optimizing features through Extra tree classifiers: The largest data set with original features selection is optimized into features subset through Extra tree classifiers which is imported from sklearn library. The feature subset obtains through Extra tree classifier.

Extra Tree Classifier: Extra tree classifier is a type of ensemble learning method which is similar to the Random Forest. The multiple decision trees are build through sample training dataset and selects the randomized subset at each node which results in optimization of features. Each Decision Tree from Extra Trees Forest is constructed from the original training sample. Then, at each test node, Each tree is provided with a random sample of k features from the featureset through which each of the decision tree must select the best feature to split the data based on some mathematical criteria (typically the Gini Index). This random sample of the features leads to creation of the de-correlated decision trees. To perform feature selection using the above forest structure, during the construction of the forest, for each feature, the normalized total reduction in the mathematical criteria used in the decision of feature of spliting (Gini Index if the Gini Index is used in the construction of the forest) is computed. For performance of feature selection, each feature is ordered in descending order according to the Gini Importance of each feature and user selects top k features according to the choice.

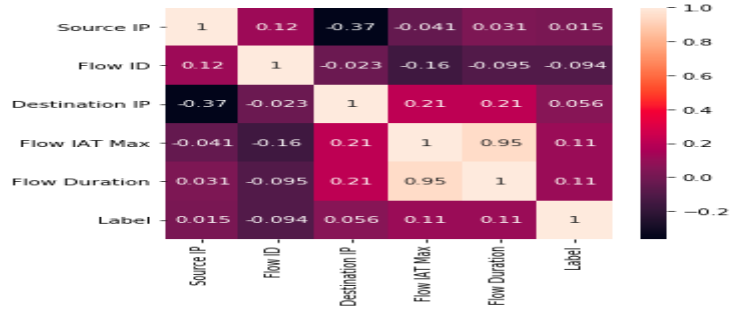
	Source IP	Flow ID	Destination IP	Flow Duration	Source Port	Label
0	0	16	9	209484	58063	1
1	0	16	9	31308	58063	1
2	0	308	32	13333	36864	1
3	13	308	0	45	443	1
4	0	500	92	22164	34482	1
...	...	...	...	...	...	...
95	1	617	130	3049110	45177	9
96	1	616	130	6029723	45176	9
97	1	617	130	5913942	45177	9
98	1	462	79	6039477	39251	9
99	1	328	35	34375	45299	9

1000 rows × 6 columns

The correlation of important features obtained through Extra tree classifier is observed.

```
plt.figure(figsize=(5,5))
sns.heatmap(sub_df.corr(),annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1868f00a630>
```



Splitting the Dataset: The optimized features are splitted into two sets for training and testing , As the dataset is in matrix format the data is splitted for training and testing. For example the shape of Dataset is (1000,6), these is splitted in row wise as X_train as (800,5), y_train as (200,5), X_test as (800,) , y_test as (200,).

Building model: The model is build to predict the type of malware in android OS through machine learning algorithms.

K- Nearest Neighbours: It is non parametric algorithm used to solve both classification and regression problems.From input k training instance output is predicted as class .The prediction is done through the majority of kNN which can be

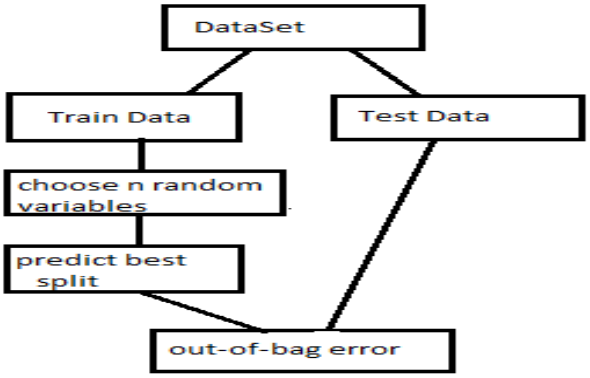
| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.089|

||Volume 9, Issue 4, April 2020||

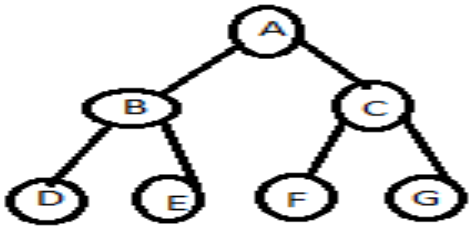
calculated through hamming distance , Manhattan Distance , Minkowski Distance. In most of the cases Euclidean distance is used which is easy to solve problems. In classification problems, odd k value is choosen which reduce complexity and ignore equal majority cases during prediction.

From above example, class of black diamond is expected to know where green and blue diamonds represents the classes. If k=3, black diamond undergoes class blue .If k=5,black diamond undergoes class green. Thus depending on k value the predictions varies.

Random Forest: Random Forest is collection of tree predictors so that each tree depends on random vector values are independently sampled with the same distribution for all trees in forest.Generalization error of a forest in Extra tree classifiers depends on strength of individual trees in the forest and the correalation between them. Features obtained from random selection is aided to split each node yields error rates that compare favorably to Adaboost, and are more robust with respect to noise.Random forest is combination of number of decision trees which obtains best results in prediction .At each node n variables are randomly choosen from training dataset and predicts best split from variables. Through test data the error is called as Out-of-bag error rate.



Decision Tree Classifier: Decision Tree is defined as control flow of different nodes and branches. Decision trees aresequential models which logically combine a sequence of simple tests where a numerical attributes is compared against a threshold values (or) a nominal attributesagainst a set of possible values. It is essentially a flow chart like structure where each internal node denotes a test of attribute and each branch represents outcome of the test with leaf holding a class label. Root node: It refers to initial input node with attributes. Decision node: It refers to internal node which makes decisions. Leaf node: Leaf node which obtains class labels through attributes. Branches: Branches are connections between different types of nodes.



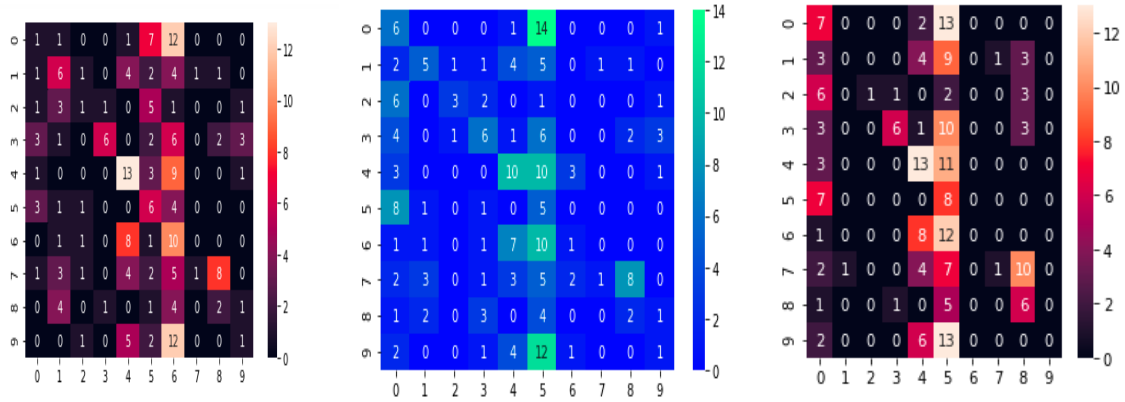
From above figure, A is root node,(B,C) are decision nodes,(D,E,F,G) are leaf nodes.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.089|

||Volume 9, Issue 4, April 2020||

V. RESULTS

To evaluate performance of proposed scheme implements detection of Android malware with optimised features. Arrangement of Android Apps or APKs: Malware are figured out to separate highlights, for example, authorizations and check of App Components, for example, Activity, Services, Content Providers, and so on. These highlights are utilized as highlight vector with class marks as Malware spoke to by 1 in CSV group. In this project using with Intel i3 and above of 64GB memory. Setup tools and pip 3.6.x and above to be installed. Anaconda 3.5 to be installed along with the different predefined libraries such as pandas,numpy, matplotlib, seaborn and scikit-learn which are need for further implementation. The libraries are installed through command prompt and using Jupyter notebook for implementation. The dataset is saved as 10 .CSV format files, the total of 1000 rows and 85 columns are used which are optimized original features into features subset .further splitting for train ,testing . Finally result is predicted. Accuracy obtained through features selected and features subset by using machine learning algorithms is in order of Random Forest > KNN > Decision Tree. The accuracy increases from features selected to feature subset obtained through Extra tree classifier.



(a) (b) (c)

(a).Confusion matrix for KNN,(b). Confusion matrix for Random Forest Classifier,(c) Confusion matrix for Decision Tree Classifier.

VI.CONCLUSION

Despite the fact that obscurity procedures speak to a boundary to identify the idea of a program, it is as yet conceivable to emerge unmistakable attributes which permit to separate practices among malicious and favorable applications. .The new model is build to detect malware with selected features. To return high accuracy rate, Extra Tree algorithm is used in this model to decrease the dimensionality of features selection. Through the obtained feature subset the accuracy rate increase for recognizing the Android malware.

REFERENCES

[1]. Naser Peiravian, Xingquan Zhu. Machine Learning For Android Malware Detection Using Permission and API calls, https://ieeexplore.ieee.org/document/6735264.
[2].SY Yerima, S Sezer. A Droidfusion: A novel multilevel classifier fusion approach for the android malware detection.
[3].Michal Kedziora, Ireneusz Jozwiak, Paulina Gawin and Michal Szczepanik. ANDROID MALWARE DETETION AND REVERSE ENGINEERING USING MACHINE LEARNING.
[4]. Balaji Baskaran, Anca Ralescu. A Study of the Android Malware Detection techniques through Machine Learning,https://www.google.com/url?sa=t&source=web&rct=j&url=http://ceur-ws.org/Vol-1584/paper19.pdf&ved=2ahUKEwjr56bT5qvoAhW763MBHRY-DzgQFjAQegQIBxAB&usg=AOvVaw0GXJq9f8Mqqt9VmHplMyZY&cshid=1584801874240.
[5].Jose Alberto Hernandez, Alfonso Munoz. Android Malware Characterization Using Metadata and Machine Learning Techniques.
[6].Mohammed k.Alzaylaee, Suleiman Y. Yerima. DL-Droid: Deep learning based android malware detection using real devices, https://www.sciencedirect.com/science/article/pii/S0167404819300016.
[7]. K. O. Elish, X. Shu, D. D. Yao, B. G. Ryder, and X. Jiang, The Profiling of user-trigger dependence for the Android malware detection, Computers & Security, vol. 49, pp. 255273,2015.