INTERNATIONAL JOURNAL OF TECHO-ENGINEERING

ISSN : 2057-5688

ANDROID MALWARE DETECTION USING EXTRA TREES CLASSIFIER BASED FEATURE SELECTION AND MACHINE LEARNING

A.TEENA BHUVANA CHANDRIKA[1], R.VIJAYANAND[2], DR.P. SRINIVASA RAO[3]

[1]M. Tech Student, Department of CSE, JB Institute of Engineering and Technology, Moinabad Mandal, Hyderabad, T.S, India

[2]Assistant Professor, Department of CSE, JB Institute of Engineering and Technology, Moinabad Mandal, Hyderabad, T.S, India

[3]Professor, Head of the Department of CSE, JB Institute of Engineering and Technology, Moinabad Mandal, Hyderabad, T.S, India

ABSTRACT:

Android platform due to open source characteristic and Google backing has the largest global market share. Being the world's most popular operating system, it has drawn the attention of cyber criminals operating particularly through wide distribution of malicious applications. This model proposes an effectual machine-learning based approach for Android Malware Detection making use of evolutionary Extra Trees Classifier for discriminatory feature selection. Selected features are used to train various machine learning classifiers such as K-Nearest Neighbors, Decision Tree, Random Forest and Support Vector Classifier with and without using hyper parameter tuning and their capability in identification of Malware before and after hyper parameter tuning is compared. The experimentation results validate that Extra Trees Classifier gives most optimized feature subset helping in reduction of feature dimension to less than half of the original feature-set. Classification accuracy of more than 70% is maintained post hyper parameter tuning, while working on much reduced feature dimension, thereby, having a positive impact on computational complexity of learning classifiers. The conclusion of the paper also states that one of the Machine Learning Classifier known as Random Forest has the greatest accuracy compared to K-Nearest Neighbors, Decision Tree and Support Vector Classifier.

Keywords: — Android Malware, Feature Selection, Extra Trees Classifier, Machine Learning, Classification.

ISSN : 2057-5688

1. INTRODUCTION:

The utilization of advanced phone has become broad now these days. Without breaking a sweat of new advances, PDAs are turning into the essential need of the end-client [1]. As Android framework is much well known, it is more powerless against malware assaults. There are many existing methodologies which are proposed by analysts for recognizing Android Malware by utilizing distinctive Machine Learning Classifiers. Malware (Malicious Software) is a general name given to any program playing out any malevolent movement. Android Apps are uninhibitedly accessible on Google Play store, the authority Android application store just as outsider application stores for clients to download. Because of its open source nature and fame, malware essayists are progressively zeroing in on creating pernicious applications for Android working framework. Notwithstanding different endeavors by Google Play store to ensure against malignant applications, they actually discover their approach to mass market and cause damage to clients by abusing individual data identified with their telephone directory, mail accounts, GPS area data and others for abuse by outsiders or probably assume liability for the telephones distantly. Subsequently, there is need to perform malware investigation or figuring out of such pernicious applications which present genuine danger to Android stages. Comprehensively speaking, Android Malware investigation is of two kinds: Static Analysis and Dynamic Analysis. Static investigation fundamentally includes dissecting the code structure without executing it while dynamic investigation is assessment of the runtime conduct of Android Apps in obliged climate. Offered in to the consistently expanding variations of Android Malware presenting zero-day dangers, a proficient system for discovery of Android malwares is required. As opposed to signature-based methodology which requires normal update of mark data set, AI based methodology in blend with static and dynamic examination can be utilized to recognize new variations of Android Malware presenting zero-day dangers. In [2], expansive yet lightweight static investigation has been performed accomplishing a respectable discovery exactness of utilizing Support Vector Machine calculation. Nikola Milosevic et al. [3] introduced static examination based

ISSN : 2057-5688

grouping through two philosophies: one was authorizations based while the other included portrayal of the source code as a pack of words. One more methodology dependent on recognizing most huge authorizations and applying AI on it for assessment has been proposed in [4]. A significant stage in all AI based methodologies is highlight choice. Acquiring ideal list of capabilities won't just assistance in further developing experimentation results however will likewise help in lessening the scourge of dimensionality related with most AI based calculations. Fest [5] proposed a novel and effective calculation for include choice to further develop by and large recognition exactness. In [6], is perspective on different element choice calculations for malware recognition has been introduced giving rules to determination? In the proposed work, Genetic calculation has been utilized due to its abilities in discovering an element subset chose from unique component vector with the end goal that it gives the best precision for classifiers on which they are prepared. It has been utilized, beforehand likewise, in mix with AI and profound learning calculations to acquire the most ideal component subset as in [7], [8]. The fundamental commitment of the work is decrease of element measurement to not exactly 50% of unique list of capabilities utilizing Genetic Algorithm to such an extent that it very well may be taken care of as contribution to AI classifiers for preparing with diminished intricacy while keeping up with their exactness in malware arrangement. Rather than comprehensive strategy for highlight choice which requires testing for 2N various mixes, where N is the quantity of components, Genetic Algorithm, a heuristic looking through approach dependent on wellness work has been utilized for include choice. The enhanced list of capabilities acquired utilizing Genetic calculation is utilized to prepare two AI calculations:

Backing Vector Machine and Neural Network. It is seen that a fair grouping precision of over kept up with while chipping away at a much lower highlight measurement ,there by, diminishing the preparation time intricacy of classifiers.

2. PROBLEM STATEMENT

Malware examination and location is an interminable contest between malware fashioners and the counter malware local area. Conventional malware identification

ISSN : 2057-5688

frameworks depend on the signature, heuristic, and cloud-based motors which can't adapt up to trendy modern malware assaults. In this way hostile to malware local area is attempting to build cutting edge Android malware identification frameworks dependent on AI and profound learning. These frameworks are created utilizing the two-venture measure (1) Feature Engineering (2) Classification. Hence we have additionally isolated the writing review into two lines of examination (1) Feature Engineering comprises of component extraction and element choice created Droid Delver (2016) for Android malware identification dependent on API calls as elements on comodo dataset proposed Droid Deep Learner, which utilized elements like authorization and API call separated from Android applications. Droid Mat removed provisions including authorization, application part, goal, and API call for building the identification model. Drebin removed consent, aim, application part, API call and organization address from the Android applications for the development of malware identification. A large portion of the above work consolidates various elements set for the improvement of Android malware identification models. For instance, Drebin utilized approximately 545, 000 unique components removed from Android applications to construct the malware identification model and subsequently experience vigorously the scourge of dimensionality.

3. PROPOSED WORK

The proposed model contains four different stages. First, preprocessing of the dataset, which means cleaning or removing white spaces in the data and conversion of categorical features into numerical values? Without preprocessing the dataset gives inaccurate data which leads to inaccurate results. After preprocessing, now the dataset is ready for the feature selection process. The second process of this proposed model is feature selection; The third process contains the supervised learning classifiers for classification. The fourth stage includes classification using hyper parameter tuning. And finally compare the classifiers in identification of malware before and after hyper parameter tuning. The Pseudo-code for the proposed model is given below.

1. Select Dataset A

ISSN : 2057-5688

2. Preprocess dataset which removes white spaces and conversion of categorical data into numerical data in the dataset A

3. Implement feature selection method on dataset A

4. Build supervised learning classifiers on selected features

5. Compute the results of classifiers

6. Impute hyper parameter tuning method on the classifiers

7. Compute the results of classifiers

8. Compare results of classifiers before and after applying hyper parameter tuning method.

The overview of proposed methodology is shown in Fig. 1and it is implemented in python script.

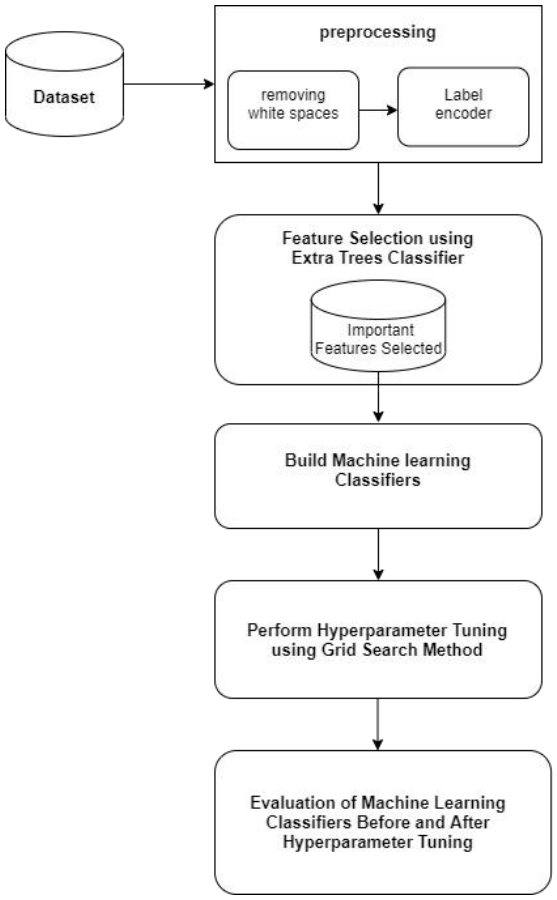

Fig. No 1. Proposed Methodology

A. Dataset Description

The dataset for this approach is collected from the Internet. The dataset contains dimensionality with 85 features and 1000 rows. Among them all files are malware files with 10 different classes.

B. Feature Selection

Feature Selection is the process of selecting the best features among total features for classification which is most relevant. The feature selection process is implemented in Python. This allows removing some features

ISSN : 2057-5688

from the dataset which are already redundant or irrelevant for the analysis. The resulting dataset usually leads to a reduced processing duration and higher accuracy compared to the raw dataset.

The preprocessed dataset is considered for feature selection. The most relevant features are selected using the feature selection method Extra Trees Classifier.

Extra Trees Classifier is an ensemble classifier used for feature selection. It is constructed using the training sample. It is a similar kind of Random Forest ensemble machine learning classifier [9]. The best features [10] are selected based on the formulae of Information Gain and Entropy which are (1) and (2) given below. It aggregates the several correlated decision trees to "forest".

$$Gain(S,A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|Sv|}{S} Entropy(Sv) \quad (1)$$

$$Entropy(S) = \sum_{i=1}^{c} - p_i \log_2(pi) \quad (2)$$

Where,

S: a training set

p_i: the proportion of rows with output label is i

c: number of unique class labels

After applying the Extra Trees Classifier feature selection method on the preprocessed dataset and it produces output as 10 best features among total 85 features with feature importance values as scores defined in Fig. 2

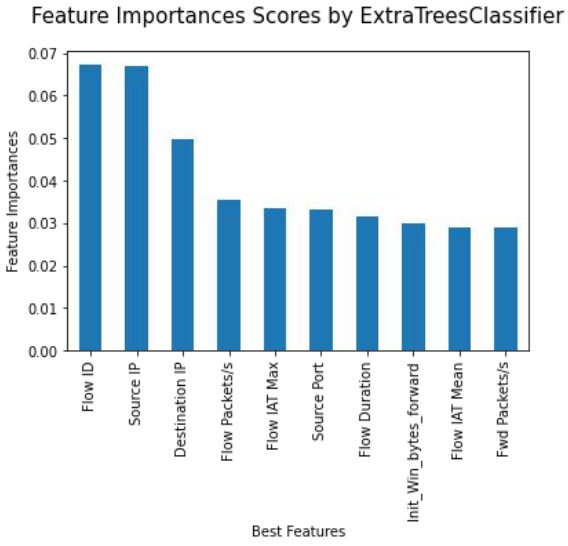

Fig.No 2.Feature Importance's of Best Selected Features by Extra Trees Classifier

C. Classification

Given in to the ever-increasing variants of Android Malware posing zero-day threat, machine learning based techniques are being preferred over traditional signature-based approach which required regular update of signature data base. The processed dataset with selected features using Extra Trees Classifier are used for classification. There are four different supervised learning

ISSN : 2057-5688

classifiers K-Nearest Neighbors, Decision Tree, Random Forest and Support Vector Classifier are used for classification in the proposed work.

D. Classification using hyper parameter tuning

Hyper parameter tuning is the process of choosing a set of optimal hyper parameters for a learning algorithm. A hyper parameter is a model argument whose value is set before the learning process begins.

A Machine Learning model is defined as a mathematical model with a number of parameters that need to be learned from the data. By training a model with existing data, we are able to fit the model parameters. However, there is another kind of parameters, known as Hyper parameters, that cannot be directly learned from the regular training process. They are usually fixed before the actual training process begins. These parameters express important properties of the model such as its complexity or how fast it should learn. Models can have many hyper parameters and finding the best combination of parameters can be treated as a search problem.

(i) Grid search

One traditional and popular way to perform hyper parameter tuning is by using an Exhaustive Grid Search from Scikit learn. This method tries every possible combination of each set of hyper-parameters. Using this method, we can find the best set of values in the parameter search space.

In Grid Search CV approach, machine learning model is evaluated for a range of hyper parameter values. This approach is called Grid Search CV, because it searches for best set of hyper parameters from a grid of hyper parameters values.

These hyper parameters are fed to different supervised learning classifiers K-Nearest Neighbors, Decision Tree, Random Forest and Support Vector Classifier for classification.

4. EXPERIMENTAL RESULTS

The results of the classification are taken under the conditions of 20% testing data and 80% Training data as it is split from the whole dataset. The experimental results of the approach are represented in the form of graph Fig. 3 which shows the performance of machine learning models before and after performing hyper parameter tuning as

ISSN : 2057-5688

accuracy.Fig.3 shows that the Random Forest classifier gives better accuracy among all other classifiers.

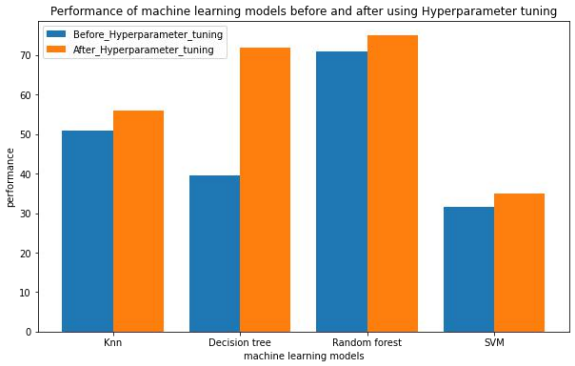

Fig. No 3.Accuracy of machine learning models before and after performing hyper parameter tuning

5. CONCLUSION

As the number of threats posed to Android platforms is increasing day to day, spreading mainly through malicious applications or malwares, therefore it is very important to design a framework which can detect such malwares with accurate results. Where signature-based approach fails to detect new variants of malware posing zero-day threats, machine learning based approaches are being used. The proposed methodology attempts to make use of Extra Tree Classifier to get the most important features which can be used to train machine learning algorithms in most efficient way. From experimentations, it can be seen that the Random Forest classifier performs better than other classifiers with accuracy of more than 75% is maintained while working on lower dimension feature-set, thereby reducing the training complexity of the classifiers. Further work can be enhanced using larger datasets for improved results and analyzing the effect on other machine learning algorithms when used in conjunction with Extra Tree Classifier.

REFERENCES:

[1]. Koli, J. D. (2018). RanDroid: Android malware detection using random machine learning classifiers. In: International Conference on Technologies for Smart City Energy Security and Power (ICSESP) IEEE, Mar 2018.

[2] D. Arp, M. Spreitzenbarth, M. Hübner, H. Gascon, and K. Rieck, "Drebin: Effective and Explainable Detection of Android Malware in Your Pocket," in Proceedings 2014 Network and Distributed System Security Symposium, 2014.

[3] N. Milosevic, A. Dehghantanha, and K. K. R. Choo, "Machine learning aided Android malware classification," Comput. Electr. Eng., vol. 61, pp. 266–274, 2017.

[4] J. Li, L. Sun, Q. Yan, Z. Li, W. Srisa-An, and H. Ye, "Significant Permission

ISSN : 2057-5688

Identification for Machine-Learning-Based Android Malware Detection," IEEE Trans. Ind. Informatics, vol. 14, no. 7, pp. 3216–3225, 2018.

[5] K. Zhao, D. Zhang, X. Su, and W. Li, "Fest : A Feature Extraction and Selection Tool for Android Malware Detection," 2015 IEEE Symp. Comput. Commun., pp. 714–720, 4893.

[6] A. Feizollah, N. B. Anuar, R. Salleh, and A. W. A. Wahab, "A review on feature selection in mobile malware detection," Digit. Investig., vol. 13, pp. 22–37, 2015.

[7] A. Martin, F. Fuentes-Hurtado, V. Naranjo, and D. Camacho, "Evolving Deep Neural Networks architectures for Android malware classification," 2017 IEEE Congr. Evol. Comput. CEC 2017 - Proc., pp. 1659–1666, 2017.

[8] A. Firdaus, N. B. Anuar, A. Karim, M. Faizal, and A. Razak, "Discovering optimal features using static analysis and a genetic search based method for Android malware detection *," vol. 19, no. 6, pp. 712– 736, 2018.

[9]. Scikit-learn (2019, December 20) [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesClassifier.html

[10]. Geeks for Geeks. (2019, December 20). A Computer Science Portal for Geeks [Online]. Available: https://www.geeksforgeeks.org/ml-extra-tree-classifier-for-feature-selection/

[11].https://www.google.com/amp/s/www.jeremyjordan.me/hyperparameter-tuning/amp/