IMPROVING CAPABILITIES OF EFFECTIVE ANDROID MALWARE DETECTION THROUGH MACHINE LEARNING TECHNIQUES

Patel Dhwani Natwarlal[1]

[1] *Student, ME (CE), Silveroak College of engineering & technology, Gujarat, India*

ABSTRACT

Now a days, drastically increased android smartphone users. Mobile apps become part of our everyday life, many of the services are provided to us through mobile apps. Therefore Android devices are the most targeted devices by malware because of their high popularity. So that it is necessary to provide Security to android device. To protect our data from malicious software, we need to detect malware from devices. Malware detection is process of detecting an unknown software and application from android device and giving security to the device. There are various ways to attack malicious program on device and previously harmful software can be detect by antivirus software but some traditional anti-virus systems are fail to classify unknown malware into their corresponding families and to detect new kinds of malware programs. So malware detection is necessary to detect malware and protect data. In this paper, we analyze the different malware detection techniques and their comparative studies. To improve malware detection accuracy, Hybrid analysis of malware and combining Boosting and decision tree algorithm used in proposed model.

Keyword: - *Android malware detection, Malware analysis, android malware, mobile security.*

1. INTRODUCTION

Android which is most common platform used across, is holding up individual's private data. Protection of data and privacy is only way by detecting malware. Malware are collecting information by breaching privacy. Malware detection methods are not effective while facing intrusions which is in trend. Malware is a short for malicious Software. Malware defined as a type of computer program designed to infect a legitimate user's computer and inflict harm on it in multiple ways. Malware is any software intentionally designed to cause damage to a computer, server or computer network. Malware (short for "malicious software") is a file or code, typically delivered over a network, that infects, explores, steals or conducts virtually any behavior an attacker wants. To have a better understanding of the methods and logic behind the malware, it is useful to classify it. Malware can be divided into several classes depending on its purpose. The classes are as follows Virus, Worms, Trojan, Spyware, Ransomware etc.

1.1 Need of Malware Detection

The increasing number of Android devices and users has been attracting the attention of different types of attackers. There are so many malicious applications troubling users due to the openness nature of the Android operating system. These malicious applications from Google Play or other third-party markets often seriously threaten users' privacy and the device's security. Thus, there are urgent needs to stop the spread of the Android malware to create a safe environment for Android smartphone users. So, if users intend to use apps from third-party vendor sites, installing Android malware protection software should be a requirement. While the diversity of malware is increasing, anti-virus scanners cannot fulfil the needs of protection, resulting in millions of hosts being attacked. Malware protection of computer systems is one of the most important cybersecurity tasks for single users and

businesses, since even a single attack can result in compromised data and sufficient losses. Massive losses and frequent attacks needs more accurate and timely detection methods.

2. MACHINE LEARNING

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine Learning is a concept which allows the machine to learn from examples and Experience. Machine Learning enables computers to get into a mode of self-learning without being explicitly programmed. When system get a new data, these computer programs are enabled to learn, change, and develop by themselves. Machine learning (ML) is the study of algorithms and mathematical models that computer systems use to progressively improve their performance on a specific task. There are several types of Machine Learning **Supervised Learning:** Supervised learning is the process of an algorithm learning from the training dataset and associated target responses in order to later predict the correct response when posed with new examples. Supervised learning is where you have input variables (x) and an output variable (Y) and you use an algorithm to learn the mapping function from the input to the output. **Unsupervised Learning:** *Unsupervised learning* occurs when an algorithm learns from plain examples without any associated response. Unsupervised learning is where you only have input data (X) and no corresponding output variables. The goal for unsupervised learning is to model the underlying structure or distribution in the data in order to learn more about the data. Unsupervised learning problems can be further grouped into clustering and association problems. **Semi-Supervised Learning:** Semi-supervised learning uses both labelled and unlabeled data for training – typically a small amount of labelled data with a large amount of unlabeled data. Problems where you have a large amount of input data (X) and only some of the data is labelled (Y) are called semi-supervised learning problems. This type of learning can be used with methods such as classification, regression and prediction. **Reinforcement Learning:** Reinforcement learning is often used for robotics, gaming and navigation. Reinforcement learning is the algorithm which discovers through trial and error which actions yield the greatest rewards [11].

2.1 Process of Machine Learning

Any machine learning task can be broken down into a series of steps:

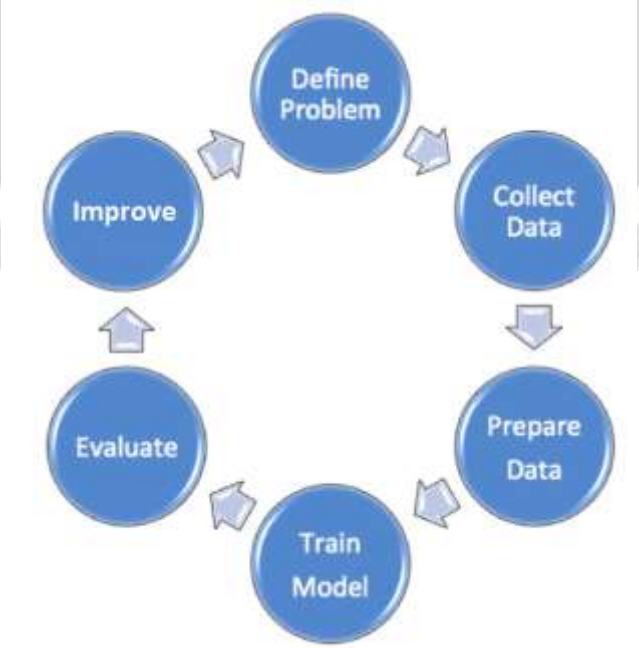


Fig.-1: Process of machine learning [12]

Selecting the Machine Leaning approach: Before starting any steps, the machine learning problem needs to be expressed. What do we want to find out? Do we want to classify our data, to predict new values, to group our data

based on some criteria? After we decide what type of machine learning task we would like to perform, we select our model.

- **Collecting data:** Data can be written on paper, recorded text files and spreadsheets or stored in an SQL database. Data need to be gathered in an electronic format suitable for analysis.
- **Exploring and preparing the data:** Data preparation, where the data is loaded into a suitable place and then prepared for use in machine learning training.
- **Training the model on the data:** The specific machine learning task will inform the selection of an appropriate algorithm. Training where the data is used to incrementally improve the model's ability to predict. The training process involves initializing some random values for say A and B of our model, predict the output with those values, then compare it with the model's prediction and then adjust the values.
- **Evaluating model performance:** It is important to evaluate how well the algorithm learned from its experience. Depending on the model used, we might be able to evaluate the accuracy of the learner using a test dataset.
- **Improving model performance:** If better performance is needed, it becomes necessary to utilize more advanced strategies to improve the performance of the model, or switch to a different model, supplement with additional data and perform additional preparation work on the data

3. REALED WORK:

Malware detection techniques are used to detect the malware and prevent the computer system from being infected by protecting it from potential information loss and system compromises. A number of techniques have been developed in order to detect malwares. These techniques can broadly be classified into two categories. Static analysis, Dynamic analysis [6] . The Static based techniques aims to analyze the malware sample without executing it. On the other hand, Dynamic based techniques analyze the malware samples by running it in a virtual environment [7][11].

Static Analysis:

This techniques are typically used to detect known malwares. The features are extracted without running the application on an emulator or device. The static analysis does not involve execution of the application. This is also referred to "Static analysis" of malware analysis. Static analysis can include various techniques: File Format Inspection, String Extraction, Fingerprinting, AV scanning, and Disassembly. Disadvantage of this technique is that it cannot detect unknown malwares. Whenever any new malware is introduced into the market it has to wait until the newly generated malwares harm several systems. Static analysis have two approaches: (a) Signature based techniques; (b) Behavior based techniques

Dynamic analysis

Dynamic analysis process includes the computation by dealing and testing of a program by executing it at real-time. The aim is to find errors in a program during the time. In this detection technique, the main focus is on discovering the malware while execution of the application. The advantage of this technique is that it analyses the behavior of the application during runtime and fetch the important data by this and name it as target data. The main purpose of behavior Based Techniques is to analyze the behavior of known or unknown malwares. The processing used for the detection of behavior in dynamic analysis includes the setting up of environment for operation followed by the simulation of the application's execution to acquire the application's behavior model. And environment setup is done by using a virtual machine, and other forms. [7]

Hybrid Analysis

Hybrid Analysis can be produced by combining both static and dynamic analysis. It is a method or technology that integrates data fetched from dynamic analysis at run-time, with static analysis algorithm for detecting the malicious behavior or suspicious functionality. This analysis method uses the static features obtained during analyzing the application combined with the dynamic features and information collected during the execution of the application. [8]

Paper 1: RanDroid: Android Malware Detection Using Random Machine Learning Classifiers. [1]

In this paper, proposed RanDroid system employs various machine learning techniques
i.e.; Support Vector Machine (SVM), Decision Tree (DT), Nave Bayes (NB) and Random Forest (RF) to perform malware classification. This paper use static analysis approach of application. The system uses permissions, API

calls along with presence of key app's information which were not considered in most of previous proposed approaches, such as: crypto code, dynamic code, native code, reflection code, and database as a features set to generate binary vector from Android application samples of identified malware and good ware applications and adopt machine learning to perform malware classification.

Paper 2: DeepDetector: Android Malware Detection using Deep Neural Network. [2]

DeepDetector shows the detail malware families and other information rather than only the result whether the applications are malicious, which makes it more convincing. We use a deep neural network consisting of more than three layers, where the inputs are feature vectors of Android applications, the middle hidden layers are parametric rectified linear unit (PReLU) activation function and the last output layer is the sigmoid unit to classify the applications. Use of Sigmoid unit to detect whether applications are malicious or not.

Paper 3: Machine learning aided Android malware classification. [3]

In this paper, we present two machine learning aided approaches for static analysis of Android malware. The first approach is based on permissions and the other is based on source code. Here use of permission names as features to build a machine learning model. Every app has to acquire the required privileges to access the different phone features. During an app installation, a user is asked whether to grant the app access to the permissions requested. Malicious apps usually require certain permissions. Propose an approach that uses the appearance of specific permissions as features for a machine learning algorithm.
The second approach is a static analysis of the app's source code. Malicious codes generally use a combination of services, methods, and API calls. In this approach, Android apps are first decompiled and then a text mining classification based on bag of- words technique is used to train the model. First, it is necessary to extract the Dalvik Executable file (dex file) from the Android application package (APK file). The second step is to transform the Dalvik Executable file into a Java archive using the dex2jar tool. After that, we extract all .class files from the Java archive and utilize Procyon Java decompiler to decompile .class files and create .java files. Then, we merge all Java source code files of the same app into one large source file. Here, use several machine learning algorithms for classifications.

Paper 4: Android Malware Detection Method Based on Function Call Graphs. [4]

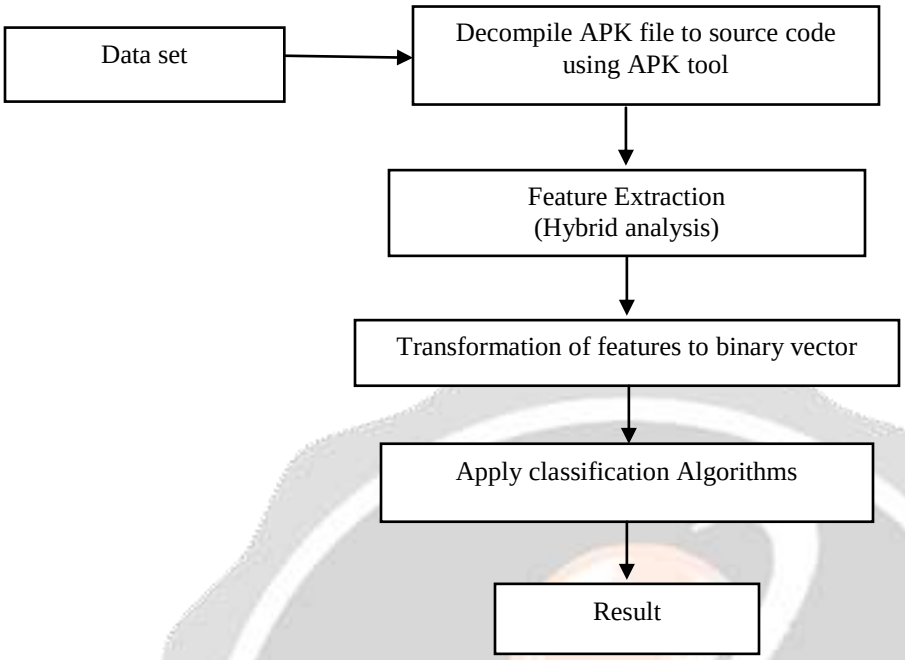
This paper focus on the static malware detection method. Static methods detect malware by analyzing the contents of each applications. In the detection model, we create three feature extraction modules which are used to extract different features. Our goal is to evaluate the performance of different features for malware detection. We extract three kinds of features: permission feature, function call feature and structure feature of function call graphs. Then decompile APK file and extract permissions used by an Android application from configure file AndroidManifest.xml. An Android application sample is represented as a permission vector in which each dimension represents a permission.

Paper 5: Malware Detection in Android based on Dynamic Analysis. [5]

We propose to analyze the behavior of malicious apps on Android using the dynamic analysis technique which is based on collecting and analyzing the frequency of system calls that are generated by applications during their run time. System call-based detection methods can detect the malicious behavior of applications more accurately than other methods since it is impossible to modify the original functionality of the system calls. This paper have employed syscall-capture system to capture and analyze the behavior of system call traces made by each application during their run time.

4. PROSOSED WORK MODEL

To find such kind of malware, machine learning techniques can be effective considering it does not need the list of known malware but works on parametric values which are the algorithmic attribute based patterns. But there is an issue regarding accuracy of malware detection model. To solve this, issue the proposed technique uses a hybrid concept of decision tree and adaptive boosting can be merged together for predicting malware and also use a hybrid technique for feature extraction of application. Here, we explain proposed model for malware detection.



Steps of Proposed System

- The proposed systems use a hybrid technique that combining static feature and dynamic features of android application to achieve a better accuracy than previous model and use effective classification algorithms to achieve an effective result. The steps of proposed system in described as follow:
- First steps of process is to upload a dataset (Android package) which user wants to check their security and detect a malicious software from data.
- Now, Decompilation process of the APK files. That is process of reverse engineering of android app to their source code in the form of androidmanifest.xml and class files of apk using APK easy tool.
- After then extract the features of application which can identify the malware availability in the files, here feature extraction is process of extracting permissions, API calls, IP addresses, Registry keys etc. from source code of the dataset are done with the sandbox tool and locally create a file that contain a features.[9]
- Now apply a transformation method to features and convert them into binary vector/ feature vector.
- Now, apply a classification algorithms on the feature vector to classify malicious application and benign application. Her we use Decision Tree concept with adaboost algorithm.

After then we get a classification result of application and calculate accuracy with bases of confusion matrix of proposed model.

5. CONCLUSIONS

There is issues related to malware detection is accuracy of the model. In proposed method we used hybrid concept of a decision tree and adaptive boosting methods for malware detecting process i.e. improved performance of system for malware detection in terms of accuracy and used hybrid analysis technique for feature extraction which is combination of static analysis and dynamic analysis of android application this also used to improve accuracy of proposed malware detection model. The main motive behind this proposed technique is to provide security to the android devices. Different malware and their types which are stealing information, encrypting or deleting sensitive

data from the android device. So that it is necessary to protect data and file from malware. For this we need to detect malware for android device.

6. REFERENCES

[1] J. D. Koli, "RanDroid: Android Malware Detection Using Random Machine Learning Classifiers", IEEE – 2018, doi: 10.1109/ICSESP.2018.8376705 .

[2] Dongfang Li1, 3, Zhaoguo Wang et al." DeepDetector: Android Malware Detection using Deep Neural Network"IEEE- 2018, doi: 10.1109/ICACCE.2018.8441737.

[3] Nikola Milosevic et al. "Machine learning aided Android malware classification" sciencedirect – 2017, doi: 10.1016/j.compeleceng.2017.02.013

[4] Yuxin Ding, Siyi Zhu, and Xiaoling Xia., "Android Malware Detection Method Based on Function Call Graphs." Springer – 2016, doi: 10.1007/978-3-319-46681-1_9

[5] Taniya Bhatia, Rishabh Kaushal. " Malware Detection in Android based on Dynamic Analysis", International Conference on Cyber Security and Protection of Digital Services – 2017, doi : 10.1109/CyberSecPODS.2017.8074847

[6] Mahima Choudhary, Brij Kishore," HAAMD: Hybrid Analysis for Android Malware Detection" IEEE-2018

[7] Manuel Egele, Theodoor Scholte, Engin Kirda, Christopher Kruegel "A Survey on Automated Dynamic Malware-Analysis Techniques and Tools", ACM Computing Surveys, Vol. 44, No. 2, Article 6, Publication date: February 2012.

[8] Abdurrahman Pekta¸s and Tankut Acarman, "Ensemble Machine Learning Approach for Android Malware Classification Using Hybrid Features" Springer International Publishing AG 2018.

[9] Chatchai Liangboonprakong, Ohm Sornil, "Classification of Malware Families Based on N-grams Sequential Pattern Features", IEEE, 2013.

[10] Fei Tonga, Zheng Yan, "A hybrid approach of mobile malware detection in Android" ,science direct- 2016, doi: http://dx.doi.org/10.1016/j.jpdc.2016.10.012

[11] Daniele Ucci., Leonardo Aniello, Roberto Baldoni "Survey of Machine Learning Techniques for Malware Analysis", Computers & Security (2018), doi: https://doi.org/10.1016/j.cose.2018.11.001.

[12] Applying Machine Learning: Steps *cs.tsu.edu/ghemri/CS696/ClassNotes/Applying%20Machine%20Learning.pdf*

[13] Machine learning *https://www.expertsystem.com/machine-learning-definition/*