Android Malware Detection Using Machine Learning Classifiers

CH.S.D.N.Durga[1], Valavala Bala Tripura Sundari[2],
[1]Assistant Professor MCA, (Ph.D) , **Dantuluri Narayana Raju College** , Bhimavaram, Andharapradesh
[2]PG Student of MCA, , **Dantuluri Narayana Raju College** , Bhimavaram, Andharapradesh

Abstract Android malware growth has been increasing dramatically along with increasing the diversity and complicity of their developing techniques. Machine learning techniques are the current methods to model patterns of static features and dynamic behaviors of Android malware. Whereas the accuracy rates of the machine learning classifiers increase with increasing the quality of the features, we relate between the apps' featuresand the features that are needed to deliver its category's functionality. Differently, our classification approach defines legitimate static features for benign apps under a specific category as opposite to identifying malicious patterns. We utilize the features of thetop rated apps in a specific category to train a malware detection classifier for thatgiven category. Android apps stores organize apps into different categories, for instance, 26 categories on Google Play Store. Each category has its distinct functionalities which means the apps under a specific category are similar in their static and dynamic features. In general, benign apps under a certain category tend to share a common set of features. On the contrary, malicious apps tend to request abnormal features, less or more than what is common for the category that they belong to. This study proposes category- based machine learning classifiers to enhance the performance of classification models at detecting malicious apps under a certain category. The intensive machine learning experiments proved that category-based classifiers report a remarkable higher average performance compared to non-category based.

Index Terms— Android malware,SVM,ANN, Google Play Store

I Introduction

According to International Data Corporation (IDC), Android OS is the most popular smartphone platform with 82.2% of the market share of smartphones, while 13.9% for iOS apple in the second quarter of 2015 [3]. Statistically speaking, it is also the first targeted platform by malware authors seeking to take the control over millions of Androidsmartphones over the world. Due to the popularity of Android's smartphones, its apps' security is a serious issue concerning 80% of smartphones users.

Android is an open source development environment that offers a rich SDK that enables developers to deploy their own apps and distribute them through Android apps centers. Android's popularity is a result of being an open source, third-party distribution

centers, a rich SDK, and the popularity of Java as

a programming language. Importantly, due to this open environment, malware authors can develop malicious apps that abuse the features that the platform offers or pack a legitimate app with a piece of malicious code; besides, exploiting vulnerabilities in the platform, hardware, or other installed apps to lunch malicious behaviors. Mainly, malware authors seek access confidential data of a device's user, monetary benefits via premium SMS, or joining the device to a botnet. Even legitimate apps introduce the risk of privacy-invading; Mcafee reported in Q1 2014 that 82% of Android apps track user's and 80% gather location data.

Research studies in the Android malware detection field work in three approaches static, dynamic or hybrid. In static analysis, malware is disassembled into a source code from where specific features are extracted. In dynamic analysis, malware is monitored at run-time in a virtual environment. In the both approaches, machine learning algorithms have been used to build classification models by training classifiers with datasets

of malware features that collected from static or dynamic analysis. The learned classification models are then used to detect malicious apps and classify them into their families.

In this study, we approach the problem differently by utilizing the features of benign apps for malware detection. We relate between the features that the app requests and the common features for its category.

Android apps stores organize apps into different categories; for example, Google play store organizes apps in 26 categories such as: "Health & Fitness", "News & Magazine", "Books & References", "Music & Audio", etc. Each category has its distinct functionalities which means the apps under a certain category share similar features. One group of these features are the permissions; per- missions are the privileges that enable apps to access the system's resources to perform their functions. Each built-in permission is responsible for providing the capabilities to execute a particular process. Apps belong to a specific category deliver the same functionality as a result they require a common combination of permissions. For instance, apps under "Communication" category commonly request READ CONTACTS but it is un- common if it is requested by apps under "News & Magazines". In general, benign apps under a certain category tend to have a common set of features: permissions, intents filters, hardware components, broadcast receivers, APIs, etc. On the contrary, malicious apps tend to request abnormal features, less or more than what is common for the category that they belong to. Repeatedly from that point of view, this study proposes category-based machine learning classifiers to enhance the performance of classification models at detecting malicious apps under a certain category. .

.

2 Literature survey

The initial studies on smart phone malware were chiefly targeted on understanding the threats behaviors of rising malware. There has been vital work on the matter of police work malware on mobile devices. Many approaches monitor the facility usage of applications and report abnormal consumption. Others monitor system calls and arrange to discover uncommon system call patterns. Different approaches additional ancient comparison with

acknowledged malware or different heuristics. Signatures primarily based ways, introduced within the mid-90s area unit ordinarily employed in malware detection. The main weakness of this kind of approach is its weakness in police work metamorphic and unseen malware. Rather than victimization predefined signatures for malware detection, data processing and machine learning techniques give a good thanks to dynamically extract malware patterns. For smart phone-based mobile computing platforms, recent years have witnessed an increasing range of additional sophisticated malware attacks like repackaging. Recent analysis consistently characterizes existing mechanical man malware from varied aspects, together with their installation ways, activation mechanism moreover because the nature of carried malicious payloads. supported the analysis with four representative mobile security software package over 1200 collected malware, their experiments show the weakness of current malware detection solutions and need the necessity to develop next-generation antimobile-malware solutions. One existing work has used data processing and options generated from windows workable API calls. They achieved sensible leads to a really giant scale dataset with concerning 35,000 transportable workable files. Another activity foot printing methodology additionally provides a dynamic approach to discover self-propagating malware. All these existing ways have basically advanced the mechanical man malware detection; however the misuse detection isn't reconciling to the novel mechanical man malware and continually needs frequent change of the signatures. Here lies the analysis gap.

In comparison, our work is motivated by a number of the higher than techniques and approaches, however with a spotlight on developing straightforward and effective malware detection approaches, while not looking forward to advanced dynamic runtime analysis and any static predefined malware signatures.

Our objective is to mix permissions and API calls as options to characterize malware and use machine learning techniques to mechanically extract patterns to differentiate benign and malicious Apps. This study may be a static analysis that uses the options that may be extracted from the supply codes of the app's .apk files

After conducting a thorough literature review of the research in the area of Android malware detection, I observed different types of objectives of such research. Many research articles are focused on surveying existing methods of solving the malware detection problem. These articles do a systematic review of different techniques that other researchers have used for this purpose and compare the results.

Liu et al. (2020) review in detail different approaches and research status from different perspectives like sample acquisition, data preprocessing, feature selection, machine learning algorithms,and performance evaluation.

Finally, Odusami et al. (2018) review existing malware detection methods, including static and dynamic analysis approaches, describe the strengths and weaknesses of each approach, and conclude that machine learning-based methods show the best detection accuracy and thus are promising for the future.

Some studies are focused on choosing the correct feature set. The features used are just as crucial to the end outcome of the malware detection exercise as the techniques and algorithms used to perform the detection. So, these studies provide valuable insight into the right feature set to use. Wenetal. (2017) use a combination of features from both static and dynamic analysis, then apply PCA to reduce the dimensionality of data and use SVM to perform the classification of applications into benign and malwareclasses. Roy et al. (2020) build a feature extraction module that performs static analysis to map each API call to certain features. Then, feature vectors are generated, and dimensionality is reduced, following which classification algorithms are used to differentiate between benign and malicious applications.

Daoudietal. (2021) convert the byte code of the application into grey-scale vector images and use 1- dimensional Convolutional Neural Networks to detect malware. This approach circumvents the need for creating comprehensive hand-crafted features and uses the raw byte code of the application for analysis. Jiang et al. (2020) study the permissions frequently used by malicious applications and identify permissions they call dangerous fine-grained permissions, which better differentiate benign and maliciousapplications. These features are then used in machine learning models to perform the classification.

Other studies focus more on optimizing the detection algorithm than the feature set. These studies are focused on improving the detection performance by choosing the right machine learning algorithm and/or using various techniques to enhance the performance of traditional algorithms. Rathore et al., 2021 use multiple types of static analysis features and compare the performance of different machine learning and deep learning techniques, both supervised and unsupervised. They found that the baseline Random Forest model without any feature reduction achieved the best performance. Shao et al., 2021 extract features from the Android application package, use the relief feature selection method to select features, use different sampling strategies to address the class imbalance, and improve traditional ensemble bagging algorithms to achieve the best performance.

There are also studies in the literature that use a combination of multiple techniques instead of applying one technique to improve results. These studies tend to develop a complex approach but can possibly lead to better results. Yerima et al. (2018) use a combination of machine learning algorithms for increased the performance of the detection system. They first perform classification using base classifiers and

then re-classify the base classifier predictions using ranking-based algorithms to achieve the final prediction. Almin et al. (2015) analyze permissions requested by applications at the time of installation and perform a combination of clustering and classification to detect malware.

In a study conducted by Syrris and Geneiatakis (2021), the authors start by appreciating how much the Android operating system has, over the years, been advancing in enhancing its robustness. The robustness is associated with the advanced technologies, significant community support, and availability of tons of resources on the internet. However, all these privileges come at a cost on source platforms in which security is compromised. In this regard, malicious applications find a way to bypass some security protocols. In addressing this problem, Syrris and Geneiatakis (2021) state that there are several approaches that can be used to leverage machine learning to detect malware through the help of static analysis data. According to Kumar et al., 2018 statistical analysis and feature extraction are the two main methodologies that support machine learning in malware detection processes.

In research that was conducted by Li et al. (2018), the authors indicated that new malicious Android applications are introduced into the mobile ecosystem every ten seconds.

This statistic is worrying, and there are chances of interfering with the mobile ecosystem growth globally if something is not done. Further, the authors acknowledge that there is a need for this problem to be addressed before it affects the integrity of the Android software engineering processes. In combating the problem, the authors acknowledge the need to have a scalable malware detection approach based on the dynamics of the mobile ecosystem and the development of Android applications.

The advancements in terms of technology in developing android operating systems have created more opportunities like the existence of e-commerce, among others. However, it has led to more challenges like cyber-attacks. Among the challenges posed by android devices and the mobile ecosystem as a whole, malicious applications have undoubtedly taken the lead (Christiana et al., 2020). The malware has also continued to advance in terms of sophistication and intelligence such that it has become hard for them to be easily detected through the existing systems. For instance, signature-based systems used for malware detection have become inefficient in detecting advanced malware applications (Christiana et al., 2020). As a result, machine learning techniques are now at the top in dealing with this challenge.

Cybercrimes are rapidly increasing on Android-based devices because of their wide usage across the globe. This increase has made it possible for malicious individuals to engineer malicious applications for their gains and at the expense of the users. In this research, the authors concur with Christiana et al. (2020) that the deployment of machine learning techniques is the option needed for curbing malicious android

applications. Malicious applications can be classified using machine learning models to differentiate between benign and malicious android applications (Sharma et al., 2020). Additionally, a comparative analysis aimed at calculating the computational time necessary to detect malicious applications is a requirement necessary in machine learning techniques.

The challenge posed by this mechanism is based on the fact that some large bundle applications cannot be easily scaled hence creating the need for Significant Permission IDentification (SigPID). This approach has an efficiency of about 93.62 percent of malware detection in a particular dataset, making it the best method to detect malware. SigPID uses permission usage to analyze the increasing number of humanoid malware. It is not necessary for the engineers to analyze all humanoid permissions for them to detect the existence of malware (Assisi et al., nd).

Instead, mining the permission information is critical in determining the most important permissions that can easily lead to the classification of malicious and benign applications, hence complementing machine learning in malware detection. Kyaw and Kham (2019) reiterate that a scalable malware detection method that yields optimum results is the use of permission analysis through the SigPID approach. Malware applications are thus identifiable through the analysis of the permission behavior. Additionally, pruning procedures are therefore necessary for identifying the most significant permissions that will provide the desired results through the multilevel pruning methodology.

Android applications are readily available because of the comprehensive community support and other open sources that make it easier for malware engineers to develop more malicious applications. Android devices have been the target for malware applications because of the worldwide reception of the android devices (Kumar et al., 2018). For the purpose of reaching high levels of accuracy when detecting malware, a small subset of specific features should be considered. Furthermore, Android is actively implementing new security controls, including the use of a unique user ID (UID) and system permissions for every application (Ranaetal., 2018). Therefore, the use of machine learning classifiers has become one of the best approaches for detecting any android malware in the mobile ecosystem and other android devices. Fallah and Bidgoly (2019) research demonstrated the need to benchmark machine learning algorithms before the associated techniques can achieve the required level of efficacy. The basis of this approach is identifying the family of particular malicious applications. Moreover, the authors demonstratethe need to use combined techniques to get optimum results that are consistent across the platforms based on the selected datasets or classifications of the malware. In this context, the authors recommend using machine learning and network-based detection techniques. In the

case of machine learning, the detection process should work in both the unsupervised and supervised machine learning methods to get viable results that will be used in the decision-making process. However, this research does not clearly demonstrate how machine learning algorithms will handle the new variants of malware that have not been tested through the existing algorithms. This means that for the machine learning techniques to be effective, the algorithms have to be updated every often to capture and detect new families of malware.

The rising attacks on smart phones result from Android being the most used OS. In this context, theauthors give a reason why it has been so easy for attackers to use malicious Android applications to launch attacks. The prompts posed to the user when installing the applications that require them to accept all sorts of necessary permissions before the installation are the main issue. Before a user gives the needed permissions, some applications fail to install, leaving the user with no option but to provide the permissions (Singh et al., 2022). Consequently, some Android applications are not approved by the associated organizations and might be misused in collecting user data that might eventually be misused. For this reason, the application of machine learning algorithms has increasingly been used in detecting Android malware. Android classification algorithms like decision trees, vector machines, and random forests form the basis of machine learning success in detecting malware on Android devices.

Considering the above types of literature available, I try to combine two types of studies in this project. The project's goal is to determine which feature set works best and explore different detection algorithms to determine which works best. It should be noted that the combination of multiple detection techniques is not in the scope of this project.

3 Implementation Study

One existing work has used data processing and options generated from windows workable API calls. They achieved sensible leads to a really giant scale dataset with concerning 35,000 transportable workable files. Another activity foot printing methodology additionally provides a dynamic approach to discover self-propagating malware. All these existing ways have basically advanced the mechanical man malware detection; however, the misuse detection isn't reconciling to the novel mechanical man malware and continually needs frequent change of the signatures.

Here lies the analysis gap. In exiting system they implemented the classifiers like naive bayes and decision tree which gives the poor accuracy

3.1proposed methodology

in the proposed system we implement a better feature extraction techniques and then we apply the genetic algorithm for feature extraction and then we use two machine learning model called

as SVM and multi perception classifier for classification of android malware detection which gives the better accuracy ratio when compare to existing system

Fig 1: Android's Stack Structure

3.2 Methodology

Modules:-

(Data Collection and Feature Filtering)

Collect all applications in separate folders which contain benign as well as suspicious applications respectively.

Using "Glob" framework in python create an array of files is for further processing.

Analyze each application in the array using "pyaxmlparser" and "androguard" framework.

Extract the following things in the analysis phase:

a. Permissions

b. Activities

c. Intents

d. API calls

2 Taking these four attributes into consideration a program maps all attributes to a CSV file and mentions a class for each application.

3 Once CSV files are generated, analyze them for any redundancy present, and if found, eliminate the entire row.

4 Another program extracts the total permissions from these APK files. These permissions will work as attributes in the Dataset CSV File (Here if permission is present it is marked as 1else it is marked as 0).

An N-bit Vector extracts search line in the CSV file, these vectors work as input to the machine learning algorithm.

4 Results and Evolution Metrics

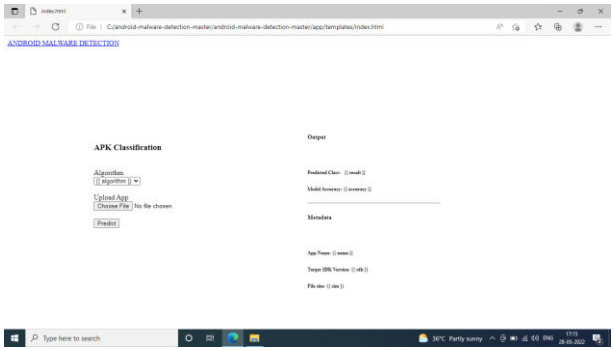
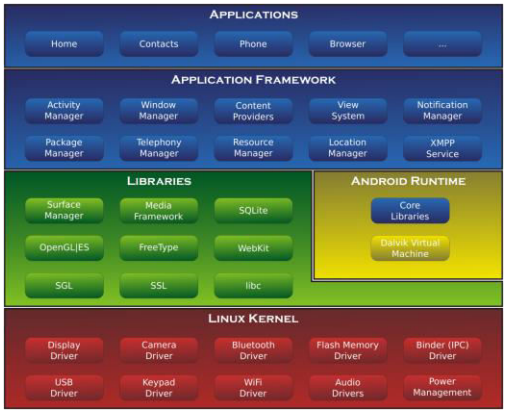


Fig 2:_ Main screen



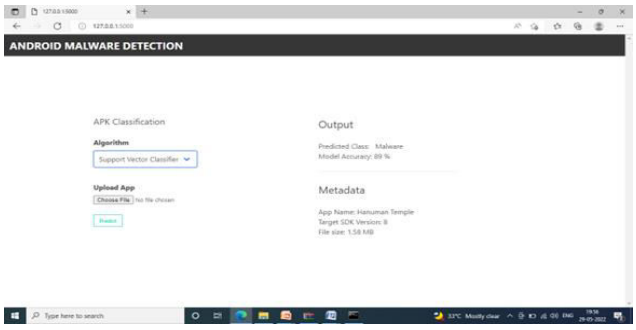


Fig 3: predicted outcome using svm

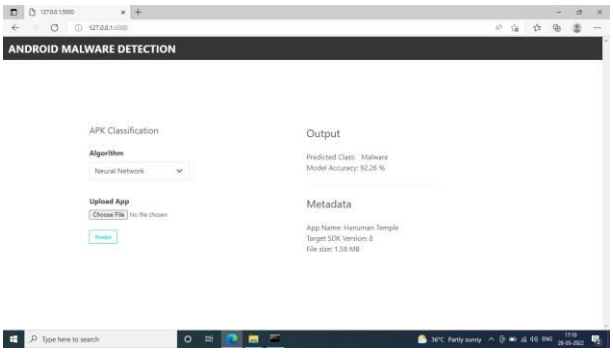


Fig 4: predicted outcome using Ann

5 Conclusion

In our study, we propose category-based machine learning classifiers to improve the performance of the classification models. In static analysis of Android malware, machine learning algorithms have been used to train classifiers with features of malicious apps to build models that capable of detecting malicious patterns. Differently, our classification approach defines legitimate static features for benign apps as opposite to identifying malicious patterns. We utilize the features of the top rated apps in a specific category to define a profile of the common sets of features for that category. In other words, to detect whether or not the app posses the characteristics of benign, we relate between the app's features and the features that are needed to deliver the category's functionality that the app belongs to. Android stores organize apps into different categories; 26 categories on the Google Play Store, for example. In each category, the apps delivera similar functionality as a result the they tend to request a common set of featureslike same permissions, APIs, hardware components, broadcast receivers, intents filters, etc. On the contrary, malicious apps tend to have abnormal features, less or more than what is common for the category that they belong to. Malicious apps can be identified by comparing between the features they request to the features that are requested by benign apps in the same category. For example, malicious apps, compared to the benign apps in the same category, tend to request over-privileged permissions, listen to specific events that broadcast by the Android system, or using unneeded APIs for the app's category functionality that can be used to lunch malicious behaviors.

6 References

[1] [Androguard usage. https://code.google.com/p/androguard/wiki/Usage. Ac- cessed April 24, 2015.

[2] Android - statistics & facts — statista. http://www.statista.com/topics/876/android/. Accessed April 19, 2015.

[3] Android and ios continue to dominate the worldwide smartphone market with an- droid shipments just shy of 800 million in 2013, according to idc. http://www.idc.com/getdoc.jsp?containerId=prUS24676414. Accessed April 19, 2015.

[4] Application fundamentals — android developers.

http://developer.android.com/ guide/components/fundamentals.html. Accessed April 19, 2015.

[5] Are — download/installation. https://redmine.honeynet.org/projects/ are/wiki. Accessed April 28, 2015.

[6] Dynamic analysis tools for android fail to detect malware with heuristic evasion techniques.http://thehackernews.com/2014/05/ dynamic-analysis-tools-for-android-fail.html. Accessed April 19, 2015.

[7] ”global smartphone sales exceed 1.2b units in 2014.” gfk - we see the big picture. http://www.gfk.com/news-and-events/press-room/press-releases/ pages/global-smartphone-sales-exceed-1-2b-units-in-2014.aspx. Accessed April 19, 2015.

[8] Google :We have 1 billion monthly activeandroid usershttp://www.businessinsider.com/ google-we-have-1-billion-monthly-active-android-users-2014-6.Accessed April 19, 2015.

[9] Report: 97- forbes. http://www.forbes.com/sites/gordonkelly/2014/03/24/ report-97-of-mobile-malware-is-on-android-this-is-the-easy-way-you-stay-safe/Accessed April 19, 2015.

•