



# ANDROID HARMFUL APPLICATIONS DETECTION USING GENETIC ALGORITHM AND MACHINE LEARNING ALGORITHMS

Mr. P. Mahesh kumar (Assistant Professor). Dept of IT Hyderabad, SNIST, Hyderabad, India

Dr. K.Kranthi Kumar ( Associate Professor), Dept of IT Hyderabad, SNIST, Hyderabad, India

Mr. G.Sumanth , Dept of IT Hyderabad, SNIST, Hyderabad, India

Mrs. P. Poojitha , Dept of IT Hyderabad, SNIST, India

Mrs. G. S.L.Sowmya, Dept of IT Hyderabad, SNIST, India

**ABSTRACT:** Android platform due to open source characteristic and android is the largest share in the global market. Being the world's most popular operating system, it has drawn the attention of cyber criminals operating particularly through wide distribution of malicious applications. In this project we propose an effective machine-learning based approach for Android Harmful Applications Detection making use of evolutionary Genetic algorithm for discriminatory feature selection. Selected features from Genetic algorithm are used to train machine learning classifiers SVM(Support Vector Machine), ANN(Artificial Neural Network), Gaussian Naïve Bayes , Random Forest, Decision Tree, XGBoost and their capability in identification of Malware before and after feature selection is compared. The experimentation results validate that Genetic algorithm gives most optimized feature subset helping in reduction of feature dimension to less than half of the original feature-set.

## 1. INTRODUCTION

Android Harmful Applications detection techniques are constantly evolving due to the necessity of detecting the presence of malware. Cyber criminals are constantly changing their techniques and novel methods of detection are needed to be developed. Moreover, Android has become one of the most popular operating systems in mobile devices. According to Static counter, Android has a market share greater than 72%. This situation has caused an increase in the malware ecosystem because of its popularity. All of this is related to the rise of smartphone users worldwide, more than 6 billion in 2021. Due to this situation, cyber criminals are increasing attacks against smartphones and the Android ecosystem in particular. On top of that, we should take into account that even in the latest Android version 11, the system still allows installing applications from unverified sources. Several malware SMS campaigns, using Smishing techniques, had exploited this possibility but the use of markets, third-party markets, and the official Google Play Store, is still the main distribution vector of infection for

most Android malware. Being Google Play Store the main distribution vector, novel techniques that control who published and which applications are published need to be developed. This evaluation is currently a challenge since there are around nearly 3 million applications in Google Play Store, making it difficult to evaluate all of them. Due to the rapid development of mobile intelligent terminals, Android becomes the most generally used computing platform on smartphones. As Trend Force (Huang Citation 2020) recently issued, a total of 1.25 billion smartphones were produced in 2020 and Android captured 78.4% of the market shares. However, due to the wide distribution and the open-source nature, Android applications are accessible from potentially malicious third parties besides the official Android Market, which makes the platform a target for malware attacks. According to the 2019 Android Malware Special Report (360 Internet Security Center Citation 2020) released by 360 Security on February 28, 2020, the platform intercepted about 1.809 million new malware samples on mobile terminals in 2019, and about 5,000 new mobile malware samples were intercepted on an average day. The rapid growth of smartphone technologies and their widespread user acceptance came simultaneously with an increase in the number and sophistication of malicious software targeting popular platforms.

Since Android OS. Driod SMS.A, the first malicious Android application was discovered in August 2010 [1], the discovery of additional Android malware has steadily increased. Based on this, the anti-malware software company Kaspersky [2] reported 5,683,694 malicious applications in 2020, the highest figure during the last 3 years. With the expected increase in the growth of malicious applications, along with the influence of Android OS, which boasted a 72.72% share globally as of May 2021 [3], it is necessary to develop a solution that can protect users by detecting malicious applications and blocking access before the damage becomes critical. To solve this problem, techniques for Android malware detection using static/dynamic analysis have emerged. As outlined by P. D. Sawle and A. B. Gadichi [4], several studies have proposed various types of methods for Android malware detection, and many different analysis tools have been suggested and utilized. Owing to the

limitations of traditional analytical techniques, detection using machine learning with static/dynamic features, and further detection through deep learning techniques, are spreading.

According to K. Liu et al. [5], many studies have applied feature selection with information gain, along with various methods for reducing the features used in machine learning, with the expectation of increasing the normalization performance and operational efficiency. These attempts are meaningful, because D. Ö., Sahin et al. [10] confirmed that feature selection could be successfully applied to detect Android malware based on the comparative experiments with various feature selection methods. However, S. Lei [11] indicated that all of these feature selection methods have certain limitations. Thus, research on applying feature selection using genetic algorithms, which are advanced in comparison to traditional methods, has emerged. Since A. Firdaus et al. [12] first introduced Android malware detection with genetic selection based on the use of a genetic algorithm, A. Fatima et al. [13] conducted a study validating its performance by building a support vector machine and neural networks with 33~40 features selected from among 99 features by applying genetic algorithms. O. Yildiz and I. A. Doğru [14] presented the experimental results of selecting 152 features from information of Android permission, choosing 16 features by applying genetic algorithm-based feature selection, and verified the performance using a decision tree, naïve Bayes, and a support vector machine. A. Meimandi et al. [15] showed a performance improvement by combining genetic algorithm and the simulated annealing with classification algorithm. J. Wang et al. [16] introduced SE droid, an Android malware detector based on a genetic algorithm and ensemble learning. L. Wang et al. [17] introduced a new algorithm based on the genetic algorithm for applications of Android malware classification problems.

## 2. LITERATURE REVIEW

### Machine learning aided Android malware classification

The widespread adoption of Android devices and their capability to access significant private and confidential information have resulted in these devices being targeted by

malware developers. Existing Android malware analysis techniques can be broadly categorized into static and dynamic analysis. 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 analysis utilizing a bag-of-words representation model. Our permission-based model is computationally inexpensive, and is implemented as the feature of OWASP Seraphim droid Android app that can be obtained from Google Play Store. Our evaluations of both approaches indicate an F-score of 95.1% and F-measure of 89% for the source code-based classification and permission-based classification models, respectively.

### **Significant Permission Identification for Machine-Learning-Based Android**

The alarming growth rate of malicious apps has become a serious issue that sets back the prosperous mobile ecosystem. A recent report indicates that a new malicious app for Android is introduced every 10 s. To combat this serious malware campaign, we need a scalable malware detection approach that can effectively and efficiently identify malware apps. Numerous malware detection tools have been developed, including system-level and network-level approaches. However, scaling the detection for a large bundle of apps remains a challenging task. In this paper, we introduce Significant Permission Identification (Sig PID), a malware detection system based on permission usage analysis to cope with the rapid increase in the number of Android malware. Instead of extracting and analyzing all Android permissions, we develop three levels of pruning by mining the permission data to identify the most significant permissions that can be effective in distinguishing between benign and malicious apps. Sig PID then utilizes machine-learning-based classification methods to classify different families of malware and benign apps. Our evaluation finds that only 22 permissions are significant. We then compare the performance of our approach, using only 22 permissions, against a baseline approach that analyzes all permissions. The results indicate that when a support vector machine is used as the classifier, we can achieve over 90% of precision, recall, accuracy, and F-measure, which are about the same as those

produced by the baseline approach while incurring the analysis times that are 4-32 times less than those of using all permissions. Compared against other state-of-the-art approaches, Sig PID is more effective by detecting 93.62% of malware in the dataset and 91.4% unknown/new malware samples.

### **2.3 A Multimodal Deep Learning Method for Android Malware Detection using Various Features**

With the widespread use of smartphones, the number of malware has been increasing exponentially. Among smart devices, android devices are the most targeted devices by malware because of their high popularity. This paper proposes a novel framework for android malware detection. Our framework uses various kinds of features to reflect the properties of android applications from various aspects, and the features are refined using our existence-based or similarity-based feature extraction method for effective feature representation on malware detection. Besides, a multimodal deep learning method is proposed to be used as a malware detection model. This paper is the first study of the multimodal deep learning to be used in the android malware detection. With our detection model, it was possible to maximize the benefits of encompassing multiple feature types. To evaluate the performance, we carried out various experiments with a total of 41 260 samples. We compared the accuracy of our model with that of other deep neural network models. Furthermore, we evaluated our framework in various aspects including the efficiency in model updates, the usefulness of diverse features, and our feature representation method. In addition, we compared the performance of our framework with those of other existing methods including deep learning-based methods.

### **3. METHODOLOGY**

Within this context, it is important to research and apply new methods of PHAs detection. Moreover, these new detection methods need to be applicable in the real world and take into account applications particularities. For example, there are devices, like Samsung, with a proprietary software development kit (SDK) that can use specific permissions like, Samsung accessory permission.

ACCESSORY\_FRAMEWORK. These permissions can only be found in certain devices and have been normalized or removed to guarantee a multiplatform market solution. It is not real to create novel detection methods that do not take into account these peculiarities or are base on unique features like C&C domains or IP addresses.

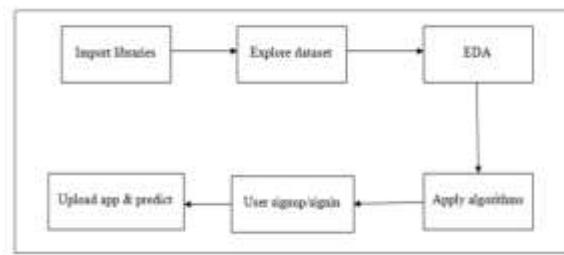
#### Disadvantages of Existing System:

1. And it is necessary to take into account that Android allows several alternative markets where there are even more malware applications.
2. Better detection rates are needed to fight malware inside application markets.
3. Furthermore, we should not forget the emergence of the Internet of Things (IOT) ecosystem and the use of Android as its operating system in these environments.

In this project proposes an effectual machine-learning based approach for Android Harmful Applications Detection making use of evolutionary Genetic algorithm for discriminatory feature selection. Selected features from Genetic algorithm are used to train machine learning classifiers SVM (Support Vector Machine), ANN (Artificial Neural Network), Gaussian Naive Bayes, Random Forest, Decision Tree, XG Boost and their capability in identification of Malware before and after feature selection is compared. The experimentation results validate that Genetic algorithm gives most optimized feature subset helping in reduction of feature dimension to less than half of the original feature-set. User sign-up & login: We will get registration and login.

#### Advantages of Proposed System:

1. Experimental results have proved that this solution obtains a 92% accuracy score
2. Our machine learning model is detecting the most common and aggressive campaigns but most elaborated ones could evade our system



**Fig.2: System architecture**

#### MODULES:

1. Data exploration
2. Processing
3. Splitting data into train & test
4. Model generation
5. User sign up & login
6. User input
7. Prediction

**Data exploration:** Using this module we will load data into system. In this data set it consists of the list of app permission of the apk file based on this permissions we can classify whether application is Harmful or Not.

**Processing:** Using the module we will read data for processing.

**Splitting data into train & test:** Using this module data will be divided into train & test.

**Model generation:** Support Vector Classifier, Artificial Neural Network, Gaussian Naive Bayes, Random Forest, Decision Tree, Stacking Classifier (Random Forest + Decision Tree) and XG boost. Algorithms accuracy calculated.

**User sign up & login:** Using this module will get registration and login. The process of registration and logging on to a site or app is the first interaction that users could have with your product. It is critical to make this step easy for the user by designing usable sign- up and login forms

User input: Using this module will give input for prediction. This the front end Page is the user interface Android malware application here the user need to sign in this application by providing the details of username, name, email id, phone number and password should enter then sign. We need to create account by the giving the user details. Here is some sample apk files we choose this apk file and upload it.

Prediction: Final predicted displayed Model Accuracy.

#### **4. IMPLEMENTATION**

## SUPPORTVECTOR CLASSIFIER:

SVC, or Support Vector Classifier, is a supervised machine learning algorithm typically used for classification tasks. SVC works by mapping data points to a high-dimensional space and then finding the optimal hyperplane that divides the data into two classes.

## ARTIFICIAL NEURAL NETWORKS:

Artificial Neural Networks (ANN) are algorithms based on brain function and are used to model complicated patterns and forecast issues. The Artificial Neural Network (ANN) is a deep learning method that arose from the concept of the human brain Biological Neural Networks.

## GAUSSIAN NAIVE BAYES:

Gaussian Naive Bayes is the extension of naive Bayes. While other functions are used to estimate data distribution, Gaussian or normal distribution is the simplest to implement as you will need to calculate the mean and standard deviation for the training data.

## RANDOM FOREST:

A Random Forest Algorithm is a supervised machine learning algorithm that is extremely popular and is used for Classification and Regression problems in Machine Learning. We know that a forest comprises numerous trees, and the more trees more it will be robust.

## DECISION TREE:

A decision tree is a non-parametric supervised learning algorithm, which is utilized for both classification and regression tasks. It has a hierarchical, tree structure, which consists of a root node, branches, internal nodes and leaf nodes.

STACKING CLASSIFIER (RANDOM FOREST + DECISION TREE):

A stacking classifier is an ensemble method where the output from multiple classifiers is passed as an input to a meta-classifier for the task of the final classification.

## 5. EXPERIMENTAL RESULTS

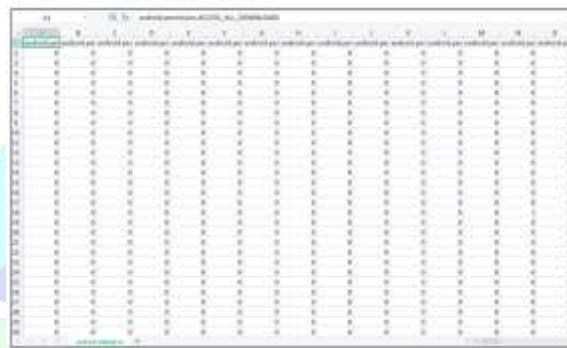


Fig.2: dataset image

In this data set it consists of the list of app permission of the apk file based on this permission's we can classify whether application is Harmful or Not. Here “0” represent no Harmful and “1” is Harmful

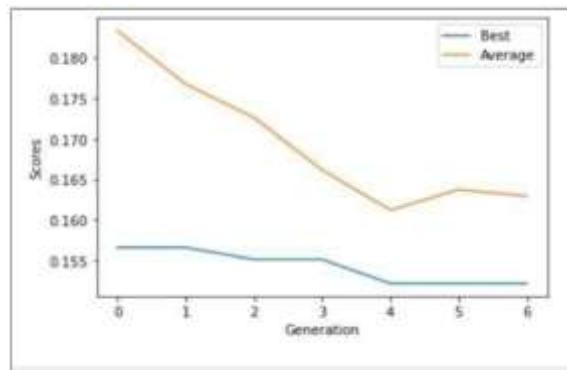


Fig.3: Generation genetic algorithm



Fig.4: User registration

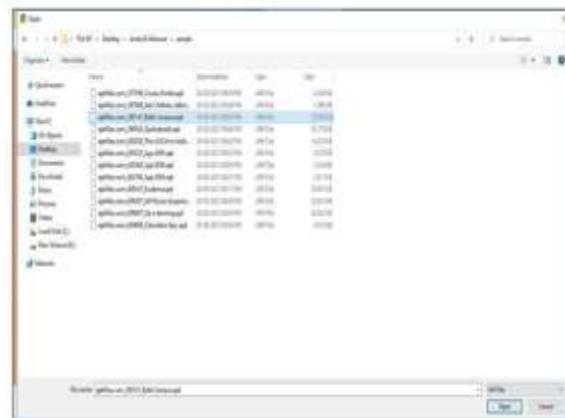


Fig.7: User input



Fig.5: User login



Fig.8: Prediction result



Fig.6: Main page

## 6. CONCLUSION

In this project presents a new way for training and detecting Potential Harmful Applications inside the Android ecosystem. The objective is to detect mobile applications that will be removed by Google in a period shorter than one month, where applications removed by Google in short periods from the store are, in most cases, Potential Harmful Applications or malware. To achieve this goal, a new data-set has been created and several classification algorithms have been used, ANN, Gaussian Naïve Bayes, Decision tree, SVM, RFC, and XGB. The data-set creation uses as criteria the lifespan of an application inside Google Play instead of antivirus decision engines, for identifying Potential Harmful Applications. Training with this data-set a Random Forest Classifier machine learning, a 92% of effectiveness can be reached.

## 7. FUTURE SCOPE

One of the main limitations of this approach is its accuracy. Future work can be done in this aspect and for example, the combination of several algorithms through ensemble learning techniques could obtain better results. Also, like any other machine learning model, it is necessary to periodically retrain this detection model with new data to detect new threats. Another possible limitation is the way that Potential Harmful Applications are selected in our datasets. The proposed approach considered Potential Harmful Applications based on the lifespan of applications inside the Google Play Store. Our selected Potential Harmful Applications are applications that Google banned or removed from the Google Play Store. But it is not possible to know how many of them were Potential Harmful Applications or applications infringing Google Play Store policies. Applications could not be a Potential Harmful Applications but Google could consider that it is infringing publishing policies.

## REFERENCES

- [1] Mobile Operating System Market Share Worldwide. Accessed: Jun. 11, 2021. [Online]. Available: <https://gs.statcounter.com/os-marketshare/mobile/worldwide>
- [2] G. Kelly. (2014). Report: 97% of Mobile Malware is on Android. This is the Easy Way You Stay Safe. [Online]. Available: <https://www.forbes.com/sites/gordonkelly/2014/03/24/report-97-of-mobile-malware-is-on-android-this-is-the-easy-way-you-stay-safe>
- [3] C. Lueg. (Jun. 2017). 8,400 New Android Malware Samples Every Day. [Online]. Available: <https://www.gdatasoftware.com/blog/2017/04/29712-8-400-new-android-malware-samples-every-day>
- [4] (2020). Smartphone Users. [Online]. Available: <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/>
- [5] J. H. Says. (Jan. 2020). SMiShing: About the FedEx SMS Phishing Scam | McAfee. [Online]. Available: </blogs/consumer/consumer-threatnotices/fedex-sms-phishing-scum/>
- [6] J. H. Says. (Jan. 2020). SMiShing: About the FedEx SMS Phishing Scam | McAfee. [Online]. Available: [/blogs/consumer/consumer-threatnotices/fedex-sms-phishing-scum/](https://blogs.consumer.consumer-threatnotices/fedex-sms-phishing-scum/)
- [7] Fake Spy Android Malware Spread Via Postal-Service Apps. Accessed: Jun. 11, 2021. [Online]. Available: <https://threatpost.com/fakespy-androidmalware-spread-via-postal-service-apps/157102/>
- [8] P. Kotzias, J. Caballero, and L. Bilge, "How did that get in my phone? Unwanted app distribution on Android devices," 2020, arXiv:2010.10088. [Online]. Available: <http://arxiv.org/abs/2010.10088>
- [9] Statista. (2017). Number of Available Applications in the Google Play Store From December 2009 to December 2020. [Online]. Available: <https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/>