# Supervised Machine Learning Framework for Fraud and Malware Detection in Android Apps

**Sujith Kumar Panda[1], Anshuman Mishra[2], Sasmita Pani[1]**

*[1]Professor, [2]Assistant Professor, [1,2]Dept. of CSE*
*[1,2]Gandhi Institute for Technology, Bhubaneshwar, India*

**Abstract**

At present, everyone is dependent upon its Smartphone for banking, communication, business, gaming and many more functionalities. But,Ransomware is one of today's most severe Internet security challenges and also Android applications also effective by the various types of Trojan attacks respectively. Indeed, most Internet issues, including spam e-mails and denial of service attacks, are triggered by malware and android applications also facing this issue. In many words, Smartphone's that are infected by Ransomware are also networked into botnets, and often assaults are performed on hostile, assaulting networks. From untrusted internet sites may be likely to contribute to maladministration. These executables are changed intelligently to circumvent antivirus specifications by anomalous users. In this article, an improved identification approach for harmful executables is suggested by evaluating Portable Executable (PE) executable files and utilizing an extraction process for support vector machine (SVM)classification. We also learned a supervised binary classifier using these features from regular and malicious PE data on Android applications. We have checked our system on a comprehensive publicly accessible dataset and obtained a rating maximum accuracy compared to the state of art approaches respectively.

**Keywords:** Machine Learning, E-mails, Networks, Malware Analysis, Feature Extraction, SVM and feature extraction.

## 1. Introduction

Usage of smart phones is increasing day by day in human life. At present, everyone is dependent upon its Smartphone for banking, communication, business, gaming and many more functionalities [1]. According to the statics at the end of 2020 ∗, there are 3.5 billion users for smart phones throughout the world. Android has gained its popularity due to its open nature and a large number (2,870,000) of apps present in its official play store at the end of March 2020 †. Due to these reasons, Android have 74.13% ffi market share and become famous in the world. By taking advantage of its open-nature, freely availability of its Android apps and its permission model, cybercriminals are developing malware-infected apps on a daily basis. By using malware-infected apps [2], cyber crooks take the personal information of users such as passwords, banking account details etc. for their benefits. According to the report published by Kaspersky §, there are 3,503,952malware packages, 68,362 Ransomware Trojans and 69,777 bankingTrojans present in Android devices [3].

The malwares can be classified into different categories according to their functionalities as follows:

- **Spyware:-**Spyware is software program that securely gathers information and sends it without user information [4].
- **Trojan:-**Trojan acts like a lawful app which can execute malevolent activities without the familiarity of the user and can steal significant details such as user passwords, credit card information etc [5].

- **Backdoor:-**Backdoor can move the manager of a instrument to an attacker without the knowledge of the owner. The attacker can perform any operations in the instrument like the owner. Thus the attacker can steal files stored in the instrument, delete files from the instrument and so on [6].
- **Worms:-**Worms can reproduce itself and spread to other instruments. It contains harmful and misleading instructions. Worms can spread through SMS/MMS messages [7].
- **Ransomware:-**Ransomware is a kind of malevolent that blocks user from access the instrument until he/she pays a ransom to the malware developers [8].
- **Root kit:-** Root kits are spiteful applications which hide the existing malware applications in the system from the normal methods of detection [9].
- **Botnet:-**Botnet is a group of grimy instrument communicating among themselves to perform MALEVOLENT activities such as DDOS attack. These instruments are infected with malware which enables the attacker to control them [10].

To solve the various types of malwares, this article is contributed as follows:

- Thisarticle majorly focuses on detecting the Trojan viruses from different websites in the smart phones and thisarticle also focused on the detection of Trojan viruses from various Android applications respectively.
- For effective classification of Malwares machine learning based classification methodology is used. the simulation results shows that the proposed method gives the outstanding performance.

## 2. METHODOLGY

The primary purpose of such a segment will be to change our previous algorithm such that malware files are correctly identified, requiring a Hundred percent success rate (where possible) on one type. Our aim throughout this paper is to examine the PE features of runtime with PCA-based enhanced function extracting for malware detection to improve protection. The PE of the runtime is a list of data items that the windows loader requires. The PE file contains different elements such as code type, application parts type, and patch count. With the support of the PE code, you will grasp how such a system should function.Design of the device in Fig. 1, the design of the scheme's two versions is seen. We have taken the information and educated the model. Upon preparation, we analyzed the .exe files, removed functionalities, and transmitted them for assessment reasons to the qualified model. The parameter which we use for specific model assessment is precision. Our design is an executable binary classifier that takes input as an executable file and outputs as an indicator that not every document is malware.

**Data source:** We have prepared our model with an accessible global data collection. The raw data includes characteristics from PE files obtained as well as deposited in a CSV file.

**PE files:** Each framework will allow everyone to insert the .exe file into the program. It removes the PE folder of both the runtime content then describes this folder in the text format. This organization of PE elements in such a text file lets us remove the appropriate functions from both the script. The services are then moved to the Function Filtration Framework for role extraction.

**Extraction feature:** such a block can be an instance from either a sample or a text layout PE script. This frame collects the characteristics from both the data input then allows the requisite preprocessing.

**ML Classifiers:** The pre-processed dataset is then divided into training and test sets. They have varying types of instruction and research samples to achieve specific outcomes. We also used optimization algorithms for machine learning and measured the reliability of the test collection.

**Model:** we save the entire best intelligence model then move the test documents on to the next. This stored pattern will enable us to assess whether or not the input file is malware. C. System modules dependent on extraction function collection using PCA.

This article also presents the general architecture of proposed malware identification plan and describes all the functionalities in detail. The figure in Figure 2 demonstrates the general structure for the detection plan of the malwares. The application of linear SVM (Support Vector Machine) approaches is discussed. Later we make use of ML classifiers to demonstrate the experimental methods and outcomes using SVM algorithm.

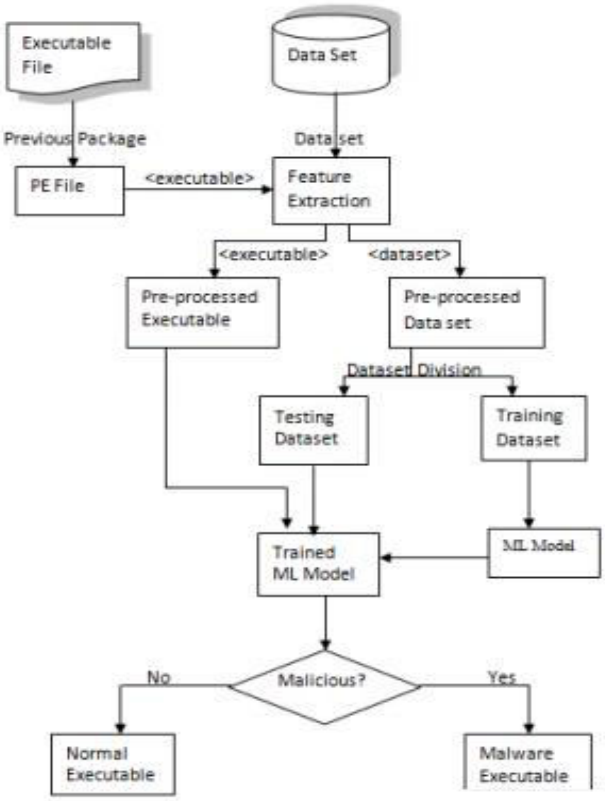


**Figure 1:** System Architecture

In the demonstration of Figure 2, there are 3 significant segments in the malware recognition conspire, to be specific decoder used to decompile, extraction of features, and classifiers. During the decompile process, the Android app unpacks and decodes intosmall files Some of the key highlights, for example, hazardous authorizations, URLs and suspicious API calls are removed in extraction parts as per a few significant and broadly acknowledged measures, for example, cosine comparability and TF-IDF. At last, we adopt ML calculation to assess on the Android application dataset by arranging them into malware or amiable applications. In this work, in order to detect the malwares linear SVM strategy is applied. The SVM is one of ML classifiers getting the most consideration right now, and its different applications are being presented due to its superior. The SVM could likewise take care of the issue of grouping nonlinear information.

From the features coming as input, pointless ones are evacuated by the SVM ML classifier and then the modeling is done, therefore there are few overheads in the part of time. Be that as it may, it could

be relied upon to obtain better results than rest of the classifiers belong to machine learning in the part of accuracy or complexity in investigation. In Our work, we select Fourteen of the recent malware apps for every classification to check the proposed technique. Malevolent applications are chosen based on the "common cases of malware making extraordinary harm to clients".

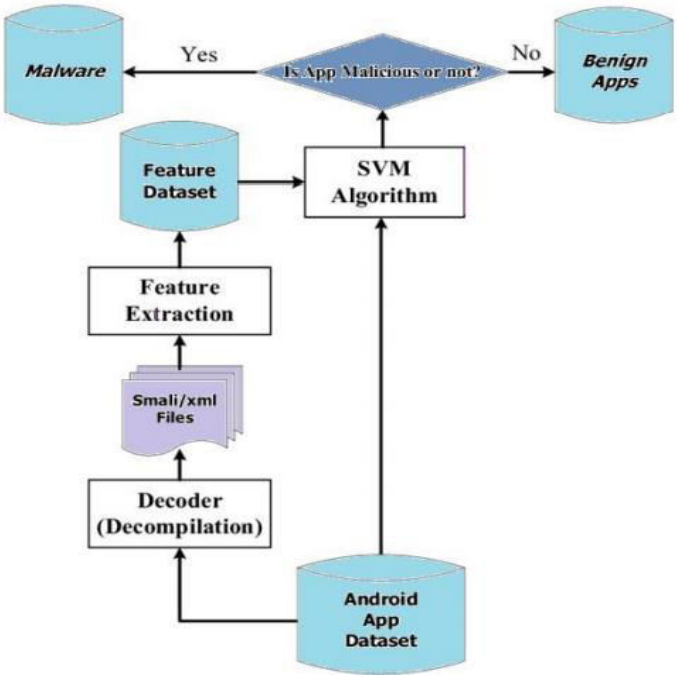


**Figure 2:** Malwares Detection in Android Smartphone's

## 3. EXPERIMENTATION AND RESUTS

The majority of the Android-focused on malwares are isolated into Spyware, Trojan, exploit, and dropper. The purpose behind Trojan having a huge extent of the chosen malware is on the grounds that the vast majority of the noxious codes that happened in 2012 were Trojan.

### 3.1 Datasets

This work utilizes total of 28 including both typical apps and malicious apps implanted with malware to verify the detection of malwares. The data collection is made out of 90% ordinary and 10% noxious apps. The purpose behind creating the informational collection thusly is that ordinary apps are more typical than noxious ones while analyzing the proportion of apps utilized in the genuine portable condition. The data is gathered from various devices in turn manner so that the gathered data is organized as the sets of training and tests. In order to test and analyze the performance belong to the experimentation we present the metrics of evaluation in this section. The metrics such as True_Positive_Rate(TPR), False_Positive_Rate (FPR), F_Measure, Precision and Accuracy are used. The True_Positive (TP) represents the numerical estimation of distinguishing the uninfected state of a typical app. The True_Negative (TN) speaks to a The TPR indicates the ratio of rightly recognized ordinary apps. The FPR indicates the ration of malwares involving apps erroneously recognized as safe. The Precision indicates the representation of an error of selection value, that speaks the ratio of rightly analyzed ordinary apps. The accuracy indicates the accuracy of the system, represented as the ratio of rightly recognized ordinary apps and ones having malwares, respectively, from the outcomes. The F_measure indicates the accuracy in the part of decision outcomes.

### 3.2 Performance evaluation

The Table 1 displays the listing of values got from the results of the malware detection using the various algorithm of machine learning technology. The computed values of Accuracy, Precision, Recall and F1 Score measures are calculated for 3different types of malware attackssuch as IRC Attack, IRC Attack and Spam Attacks along with the normalized values of the same.

**Table 1:** Results of the malware detection based on the various classifiers

| (a) IRC Attack | | | | |
|---|---|---|---|---|
| Algorithm | Accuracy | Precision | Recall | F1 Score |
| Decision Tree [12] | 0.87 | 0.68 | - | 0.77 |
| Proposed SVM | 0.9831 | 0.9665 | 0.9831 | 0.9747 |
| (b) DDoS Attack | | | | |
| Algorithm | Accuracy | Precision | Recall | F1 Score |
| Decision Tree [12] | 0.81 | 0.94 | - | 0.33 |
| Proposed SVM | 0.9931 | 0.9929 | 0.9931 | 0.9929 |
| (c) Spam Attack | | | | |
| Algorithm | Accuracy | Precision | Recall | F1 Score |
| Decision Tree [12] | 0.948 | 0.925 | - | 0.923 |
| Proposed SVM | 0.9839 | 0.9680 | 0.9839 | 0.9759 |

Among the perspective of Accuracy (0.999), the support vector machine gives a better performance. The metric FPR is utilized as the most significant assessment marker when identifying malware, The SVM achieves FPR = 0.004, that can be resolved as the better classifier since its proportion of erroneously arranging ordinary applications as malignant is little, and it appears far superior execution than different classifiers likewise as far as precision and accuracy.

**Table 2:** Training and Testing Time comparison

| Methods | Accuracy (%) | Training time (minutes) | Detection time (seconds) |
|---|---|---|---|
| Decision Tree [12] | 97.58 | 13.05 | 5.56 |
| Proposed SVM | 99.47 | 3.01 | 0.03 |

From the Table 2, it is observed that the proposed method consumes less time for Training of the network and also compares the less time for detection of Malwares with highest accuracy compared to the state of art Decision Tree [12].

## 4. CONCLUSION

The work done above proposes the mechanism for detecting the malwares based on SVM methodology for Android systems. The concept makes use of combination of risky authorizations and unsafe API calls as highlights to develop the SVM classifications that can naturally recognize vindictive Android applications from genuine ones. The results of the analysis prove that the presented plan can recognize malware in a precise way. We inferred that SVM method would precisely distinguish much of the malwares from a relative perspective by nearly breaking down them. The future investigations will consider other classifications as well, uncovering scarcely noticeable malware by resource data and more honed framework precision.

**References**

[1]. Rahman, Mahmudur, et al. "Search rank fraud and malware detection in Google Play." *IEEE Transactions on Knowledge and Data Engineering* 29.6 (2017): 1329-1342.

[2]. Rahman, Mahmudur, et al. "Fairplay: Fraud and malware detection in google play." *Proceedings of the 2016 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, 2016.

[3]. Alazab, Moutaz, et al. "Intelligent mobile malware detection using permission requests and API calls." *Future Generation Computer Systems* 107 (2020): 509-521.

[4]. Shabtai, Asaf, et al. ""“Andromaly”: a behavioral malware detection framework for android devices." *Journal of Intelligent Information Systems* 38.1 (2012): 161-190.

[5]. Mane, Ashwini Kidile1 Shweta Jadhav2 Amruta, and Sushant Borate4 Kalpana Kadam. "A Review on Fraud and Malware Detection in Google Play."

[6]. Asha, P., T. Lahari, and B. Kavya. "Comprehensive Behaviour of Malware Detection Using the Machine Learning Classifier." *International Conference on Soft Computing Systems*. Springer, Singapore, 2018.

[7]. MOUNIKA, A., and D. PREM KUMAR. "Malware Detection in Web Application using Content Integrity Verification." *International Journal of Recent Trends in Engineering and Research* 4.3 (2018): 460-464.

[8]. Seraj, Saeed, Michalis Pavlidis, and Nikolaos Polatidis. "A novel dataset for fake android anti-malware detection." *Proceedings of the 10th International Conference on Web Intelligence, Mining and Semantics*. 2020.

[9]. Firdaus, Ahmad, et al. "Discovering optimal features using static analysis and a genetic search based method for Android malware detection." *Frontiers of Information Technology & Electronic Engineering* 19.6 (2018): 712-736.

[10]. Han, Qian, V. S. Subrahmanian, and Yanhai Xiong. "Android Malware Detection via (Somewhat) Robust Irreversible Feature Transformations." *IEEE Transactions on Information Forensics and Security* 15 (2020): 3511-3525.

[11]. Saif, Dina, S. M. El-Gokhy, and E. Sallam. "Deep Belief Networks-based framework for malware detection in Android systems." *Alexandria engineering journal* 57.4 (2018): 4049-4057.

[12]. Dasari, Deepika, M. Kameswara Rao, and Nikhitha Namburu. "A Novel Mechanism for Fraud Rank Detection in Social Networks." *Inventive Communication and Computational Technologies*. Springer, Singapore, 2021. 519-526.