

## **MALWARE DETECTION : A FRAMEWORK FOR REVERSE ENGINEERED ANDROID APPLICATIONS THROUGH MACHINE LEARNING ALGORITHMS**

**<sup>1</sup>SANAKOUSAR MAHAMAD, <sup>2</sup>B PAVAN KUMAR GOUD, <sup>3</sup>K SRAVANI,  
<sup>4</sup>KETHAVATH MAMATHA**

<sup>123</sup>ASSISTANCT PROFESSOR, BRILLIANT INSTITUTE OF ENGINEERING &  
TECHNOLOGY, ABDULLAPURMET(V&M) RANGA REDDY DIST-501505

<sup>4</sup>UG SCHOLAR, DEPARTMENT OF CSE, BRILLIANT INSTITUTE OF ENGINEERING  
& TECHNOLOGY, ABDULLAPURMET(V&M) RANGA REDDY DIST-501505

### **ABSTRACT**

Today, Android is one of the most used operating systems in smartphone technology. This is the main reason, Android has become the favorite target for hackers and attackers. Malicious codes are being embedded in Android applications in such a sophisticated manner that detecting and identifying an application as a malware has become the toughest job for security providers. In terms of ingenuity and cognition, Android malware has progressed to the point where they're more impervious to conventional detection techniques. Approaches based on machine learning have emerged as a much more effective way to tackle the intricacy and originality of developing Android threats. They function by first identifying current patterns of malware activity and then using this information to distinguish between identified threats and unidentified threats with unknown behavior. This research paper uses Reverse Engineered Android applications' features and Machine Learning algorithms to find vulnerabilities present in Smartphone applications. Our contribution is twofold. Firstly, we propose a model that incorporates more innovative static feature sets with the largest current datasets of malware samples than conventional methods. Secondly, we have used ensemble learning with machine learning algorithms such as AdaBoost, SVM, etc. to improve our model's performance. Our experimental results and findings exhibit 96.24% accuracy to detect extracted malware from Android applications, with a 0.3 False Positive Rate (FPR). The proposed model incorporates ignored detrimental features such as permissions, intents, API calls, and so on, trained by feeding a solitary arbitrary feature, extracted by reverse engineering as an input to the machine.

### **I. INTRODUCTION**

The project stems from the escalating threat of malicious Android applications that compromise user security and privacy. With the rapid evolution of sophisticated malware, traditional detection methods prove insufficient. Recognizing the value of reverse engineering in understanding malicious software, this project aims to develop a robust framework integrating

machine learning algorithms for the automated analysis of reverse-engineered Android applications. By leveraging machine learning, the project seeks to enhance the accuracy and efficiency of malware detection, contributing to a more secure and resilient mobile ecosystem.

### **PROBLEM DEFINITION**

The problem at the core of this endeavor is the detection of Android malware, a



challenge exacerbated by the limitations of traditional signature-based methods in keeping pace with evolving threats. Reverse engineering offers a valuable means to comprehend application internals, yet the analysis process is intricate and time-consuming. The project addresses this problem by proposing a framework that automates the analysis of reverse-engineered Android applications using machine learning algorithms. The goal is to develop models capable of discerning between benign and malicious behaviors, identifying novel threats, and establishing a proactive defense against emerging security risks.

## OBJECTIVE OF PROJECT

The primary objective of the project is the creation of a comprehensive framework for detecting malware in Android applications through the application of machine learning algorithms to reverse-engineered data. This involves developing automated tools for reverse engineering, implementing machine learning models for distinguishing between benign and malicious patterns, focusing on behavioral analysis to enhance detection capabilities, providing real-time malware detection during application installation or execution, and ensuring adaptability to new and emerging threats through continuous updates and model training.

## SCOPE OF PROJECT

The scope of the project encompasses several key areas, including in-depth reverse engineering analysis, feature extraction from reverse-engineered data, training of machine learning models, real-time detection integration, scalability to handle diverse applications, and user-friendly design for adoption by security professionals and developers. By addressing these objectives

within the defined scope, the project aims to advance the field of Android malware detection, contributing to improved security practices for mobile devices.

## II.EXISTING SYSTEM

The methods proposed in this related work contribute to key aspects and a higher predictive rate for malware detection. Certain research has focused on increasing accuracy, while others have focused on providing a larger dataset, some have been implemented by employing various feature sets, and many studies have combined all of these to improve detection rate efficiency. In [21] the authors offer a system for detecting Android malware apps to aid in the organization of the Android Market. The proposed framework aims to provide a machine learning-based malware detection system for Android to detect malware apps and improve phone users' safety and privacy. This system monitors different permission-based characteristics and events acquired from Android apps and examines these features employing machine learning classifiers to determine if the program is goodware or malicious.

The paper uses two datasets with collectively 700 malware samples and 160 features. Both datasets achieved approximately 91% accuracy with Random Forest (RF) Algorithm. [22] Examines 5,560 malware samples, detecting 94 % of the malware with minimal false alarms, where the reasons supplied for each detection disclose key features of the identified malware. Another technique [23] exceeds both static and dynamic methods that rely on system calls in terms of resilience. Researchers demonstrated the consistency



of the model in attaining maximum classification performance and better accuracy compared to two state-of-the-art peer methods that represent both static and dynamic methodologies over for nine years through three interrelated assessments with satisfactory malware samples from different sources. Model continuously achieved 97% F1- measure accuracy for identifying applications or categorizing malware.

[24] The authors present a unique Android malware detection approach dubbed Permission- based Malware Detection Systems (PMDS) based on a study of 2950 samples of benign and malicious Android applications. In PMDS, requested permissions are viewed as behavioral markers, and a machine learning model is built on those indicators to detect new potentially dangerous behavior in unknown apps depending on the mix of rights they require. PMDS identifies more than 92–94% of all heretofore unknown malware, with a false positive rate of 1.52–3.93%.

The authors of this article [25] solely use the machine learning ensemble learning method Random Forest supervised classifier on Android feature malware samples with 42 features respectively. Their objective was to assess Random Forest's accuracy in identifying Android application activity as harmful or benign. Dataset 1 is built on 1330 malicious apk samples and 407 benign ones seen by the author. This is based on the collection of feature vectors for each application. Based on an ensemble learning approach, Congyi proposes a concept in [26] for recognizing and distinguishing Android malware.

## Disadvantages

- ❖ The system is not implemented MACHINE LEARNING ALGORITHM AND ENSEMBLE LEARNING.
- ❖ The system is not implemented Reverse Engineered Applications characteristics.

## III.PROPOSED SYSTEM

1) We present a novel subset of features for static detection of Android malware, which consists of seven additional selected feature sets that are using around 56000 features from these categories. On a collection of more than 500k benign and malicious Android applications and the highest malware sample set than any state-of-the-art approach, we assess their stability. The results obtain a detection increase in accuracy to 96.24 % with 0.3% false-positives.

2) With the additional features, we have trained six classifier models or machine learning algorithms and also implemented a Boosting ensemble learning approach (AdaBoost) with a Decision Tree based on the binary classification to enhance our prediction rate. 3) Our model is trained on the latest and large time aware samples of malware collected within recent years including the latest Android API level than state-of-the-art approaches.

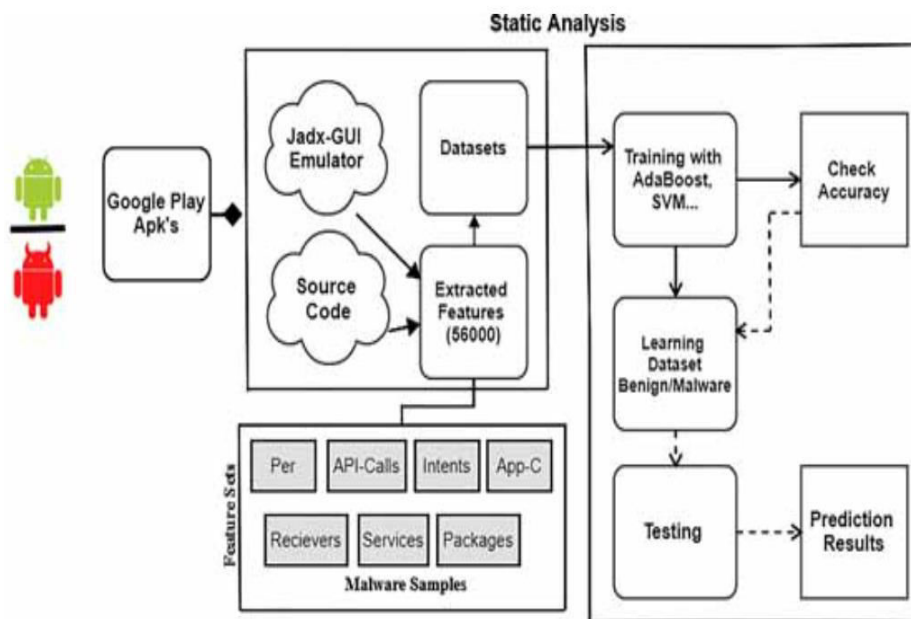
## Advantages

- The proposed system chooses the characteristics based on their capability to display all data sets. Enhanced efficiency by reducing the

dataset size and the hours wasted on the classification process introduces an effective function selection process.

- The system used in this study also incorporates larger feature sets for classification.

Although this problem arises in machine learning quite often to some extent choosing the type of model for detection or classification can highly impact the high dimensionality of the data being used.



#### IV. MODULES:

**1.Data Collection Module:** The Data Collection Module serves as the foundation for the framework by gathering information from reverse-engineered Android applications. It conducts both static and dynamic analyses during application execution, extracting relevant features crucial for subsequent stages. This module ensures that comprehensive data is collected for effective analysis by the system.

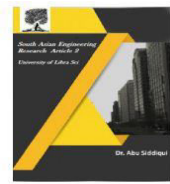
**2.Feature Extraction Module:** The Feature Extraction Module is responsible for identifying and extracting pertinent features from the collected data. It analyzes static features such as permissions requested and

API calls, along with processing dynamic features like runtime behavior and system interactions. The module converts raw data into feature vectors, preparing the input for the machine learning models.

**3.Machine Learning Module:** The Machine Learning Module is at the core of the framework, implementing machine learning algorithms for effective malware detection. It involves training models using labeled datasets, validating their accuracy, and integrating them into the system for real-time detection during application installation or execution. This module is crucial for enhancing the system's ability to discern between benign and malicious behaviors.



2581-4575



**4.Real-Time Detection Module:** The Real-Time Detection Module ensures the constant monitoring and analysis of applications during installation or execution. It triggers the machine learning models for on-the-fly detection, comparing application behavior against known malware patterns. This module plays a vital role in providing timely alerts and taking preventive actions upon detecting potential threats.

**5.User Interface (UI) Module:** The User Interface Module provides a user-friendly front-end for system interaction. It displays relevant information on detected malware, allowing users to initiate scans or customize detection settings. Clear alerts and recommendations are presented through the interface, ensuring effective communication between the system and users.

**6.Database Management Module:** The Database Management Module handles the storage and retrieval of datasets and model parameters. It manages historical data for analysis and model training, ensuring efficient retrieval for real-time detection. This module is crucial for maintaining a well-organized and accessible repository of information.

**7.Reporting and Logging Module:** The Reporting and Logging Module generates reports and logs for analysis and user awareness. It records detection outcomes and reasons, providing detailed reports on detected malware. System administrators and users can access logs to gain insights into the system's activities, contributing to transparency and accountability.

## V.CONCLUSION

In conclusion, the proposed framework for Android malware detection represents a

significant advancement in addressing the challenges posed by the existing methods. By leveraging machine learning algorithms and prioritizing behavioral analysis, the system introduces adaptability, learning capabilities, and a proactive defense mechanism against emerging threats. The emphasis on real-time detection during application installation or execution enhances responsiveness, ensuring the timely identification and mitigation of malicious activities.

The scalability of the proposed system, achieved through the reduction of manual efforts in signature creation and rule definition, facilitates the handling of a growing dataset of diverse Android applications. The continuous learning mechanism guarantees adaptability to evolving threats, maintaining a high level of detection accuracy over time. The system's focus on minimizing false positives, aided by nuanced feature extraction and analysis through machine learning, contributes to optimized resource utilization and improved overall performance.

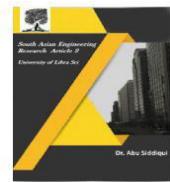
In essence, the proposed framework offers a comprehensive and forward-thinking solution for Android malware detection, striving to elevate the effectiveness, adaptability, and efficiency of the detection process within the dynamic landscape of mobile security. Through these innovations, the project aims to contribute significantly to the enhancement of Android application security, ensuring the safety and privacy of mobile device users in the face of evolving cybersecurity challenges.

## VI.REFERENCES

*1.Android (GOOG) Just Hit a Record 88% Market Share of All Smartphones—Quartz,*



2581-4575



- Jan. 2022, [online] Available: <https://qz.com/826672/android-goog-just-hit-a-record-88-market-share-of-all-smartphones/>.
- 2.A. O. Christiana, B. A. Gyunka and A. Noah, "Android malware detection through machine learning techniques: A review", *Int. J. Online Biomed. Eng.*, vol. 16, no. 2, pp. 14, Feb. 2020.
- 3.D. Ghimire and J. Lee, "Geometric feature-based facial expression recognition in image sequences using multi-class AdaBoost and support vector machines", *Sensors*, vol. 13, no. 6, pp. 7714-7734, Jun. 2013.
- 4.R. Wang, "AdaBoost for feature selection classification and its relation with SVM a review", *Phys. Proc.*, vol. 25, pp. 800-807, Jan. 2012.
- 5.J. Sun, H. Fujita, P. Chen and H. Li, "Dynamic financial distress prediction with concept drift based on time weighting combined with Adaboost support vector machine ensemble", *Knowl.-Based Syst.*, vol. 120, pp. 4-14, Mar. 2017.
- 6.A. Garg and K. Tai, "Comparison of statistical and machine learning methods in modelling of data with multicollinearity", *Int. J. Model. Identificat. Control*, vol. 18, no. 4, pp. 295, 2013.
- 7.C. P. Obite, N. P. Olewuezi, G. U. Ugwuanyim and D. C. Bartholomew, "Multicollinearity effect in regression analysis: A feed forward artificial neural network approach", *Asian J. Probab. Statist.*, vol. 6, no. 1, pp. 22-33, Jan. 2020.
8. W. Wang, M. Zhao, Z. Gao, G. Xu, H. Xian, Y. Li, et al., "Constructing features for detecting Android malicious applications: Issues taxonomy and directions", *IEEE Access*, vol. 7, pp. 67602-67631, 2019.
- 9.B. Rashidi, C. Fung and E. Bertino, "Android malicious application detection using support vector machine and active learning", *Proc. 13th Int. Conf. Netw. Service Manage. (CNSM)*, pp. 1-9, Nov. 2017.
- 10.J. Li, L. Sun, Q. Yan, Z. Li, W. Srisa-An and H. Ye, "Significant permission identification for machine-learning-based Android malware detection", *IEEE Trans. Ind. Informat.*, vol. 14, no. 7, pp. 3216-3225, Jul. 2018.
- 11.G. Suarez-Tangil, J. E. Tapiador, P. Peris-Lopez and J. Blasco, "Dendroid: A text mining approach to analyzing and classifying code structures in Android malware families", *Exp. Syst. Appl.*, vol. 41, no. 4, pp. 1104-1117, Mar. 2014.
- 12.M. Magdum, "Permission based mobile malware detection system using machine learning", *Techniques*, vol. 14, no. 6, pp. 6170-6174, 2015.
- 13.M. Qiao, A. H. Sung and Q. Liu, "Merging permission and API features for Android malware detection", *Proc. 5th IIAI Int. Congr. Adv. Appl. Informat. (IIAI-AAI)*, pp. 566-571, Jul. 2016.
- 14.D. O. Sahin, O. E. Kural, S. Akleylek and E. Kilic, "New results on permission based static analysis for Android malware", *Proc. 6th Int. Symp. Digit. Forensic Secur. (ISDFS)*, pp. 1-4, Mar. 2018.
15. A. Mahindru and A. L. Sangal, "MLDroid—Framework for Android malware detection using machine learning techniques", *Neural Comput. Appl.*, vol. 33, no. 10, pp. 5183-5240, May 2021.
- 16.X. Su, D. Zhang, W. Li and K. Zhao, "A deep learning approach to Android malware feature learning and detection", *Proc. IEEE Trustcom/BigDataSE/ISPA*, pp. 244-251, Aug. 2016.



2581-4575



17.K. A. Talha, D. I. Alper and C. Aydin, "APK auditor: Permission-based Android malware detection system", *Digit. Invest.*, vol. 13, pp. 1-14, Jun. 2015.

18.A. Mahindru and P. Singh, "Dynamic permissions based Android malware detection using machine learning techniques", *Proc. 10th Innov. Softw. Eng. Conf.*, pp. 202-210, Feb. 2017.

19.U. Pehlivan, N. Baltaci, C. Acarturk and N. Baykal, "The analysis of feature selection methods and classification algorithms in permission based Android malware detection", *Proc. IEEE Symp. Comput. Intell. Cyber Secur. (CICS)*, pp. 1-8, Dec. 2014.

20.M. Kedziora, P. Gawin, M. Szczepanik and I. Jozwiak, "Malware detection using machine learning algorithms and reverse engineering of Android Java code", *Int. J. Netw. Secur. Appl.*, vol. 11, no. 1, pp. 1-14, Jan. 2019.

She received B.Tech in INFORMATION TECHNOLOGY Degree from Christu Jyoti institute of technology and sciences in 2009 and received MASTER OF ENGINEERING (COMPUTER SCIENCE AND ENGINEERING) Degree from vaageswari college of engineering in 2013. She is having Academic Experience of more than 13 years. She is associated with MRECW since 10 Years. Her current area of research includes Machine Learning and Data Science. She is having 6 papers in reputed International Journals. Attended Various Workshops and Faculty Development Programs.

