International Journal of Engineering Technology and Management Science
Website: ijetms.in Issue: 2 Volume No.9 March - April – 2025
DOI:10.46647/ijetms.2025.v09i02.093 ISSN: 2581-4621

Malware Detection Technique for Android Devices Using Machine Learning Algorithms

UMD Azran Ali[1], CH. Sri Lakshmi Prasanna[2]
[1]*MCA Student, Dr.K.V.Subba Reddy Institute of Technology, Kurnool, Andhra Pradesh, India*
[2]*Assistant Professor, Dr.K.V.Subba Reddy Institute of Technology, Kurnool, Andhra Pradesh, India*

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.
Keywords— Smart phones , Malware , Feature extraction , Machine learning algorithms, Predictive models , Machine learning , Static analysis.

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.

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.

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

DOI:10.46647/ijetms.2025.v09i02.093 ISSN: 2581-4621

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.

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.

Literature Survey

The Literature review plays a very important role in the research process. It is a source fromhere research ideas are drawn and developed into concepts and finally theories. It also providesthe researchers a bird's eye view about the research done in that area so far. Depending on whatis observed in the literature review, a researcher will understand where his/her research stands.Here in this literature survey, all primary, secondary and tertiary sources of information weresearched. A literature survey or literature review means that researcher read and report on whatthe literature in the field has to say about the topic or subject. It is a study and review of relevantliterature materials in relation to a topic that have been given.

1. Title :Android malware detection through machine learning techniques Author :A Nova and A Guyunka Description :With the increasing use of mobile devices, malware attacks are rising, especiallyon Android phones, which account for 72.2% of the total market share. Hackers try to attacksmartphones with various methods such as credential theft, surveillance, and maliciousadvertising. Among numerous countermeasures, machine learning (ML)-based methods haveproven to be an effective means of detecting these attacks, as they are able to derive a classifierfrom a set of training examples, thus eliminating the need for an explicit definition of thesignatures when developing malware detectors. This paper provides a systematic review of MLbasedAndroid malware detection techniques.

2.Title :Geometric feature-based facial expression recognition in image sequences using multiclassAdaBoost and support vector machines

Author :J Lee and D Gambhire

Description :Facial expressions are widely used in the behavioral interpretation of emotions,cognitive science, and social interactions. In this paper, present a novel method for fullyautomatic facial expression recognition in facial image sequences. As the facial expressionevolves over time facial landmarks are automatically tracked in consecutive video frames, usingdisplacements based on elastic bunch graph matching displacement estimation. Feature vectorsfrom individual landmarks, as well as pairs of landmarks tracking results are extracted, andnormalized, with respect to the first frame in the sequence. The prototypical expression sequencefor each class of facial expression is formed, by taking the median of the landmark trackingresults from the training facial expression sequences. Multi-class AdaBoost with dynamic timewarping similarity distance between the feature vector of input facial expression and prototypicalfacial expression, is used as a weak classifier to select the subset of discriminative featurevectors. Finally, two methods for facial expression recognition are presented, either by usingmulti-class AdaBoost with dynamic time warping, or by using support vector machine on theboosted feature vectors.

3.Title :Android malicious application detection using support vector machine and activelearning

Author :C Fung and B Rashidi

Description :The increasing popularity of Android phones and its open app market system have caused the proliferation of malicious Android apps. The increasing sophistication and diversityof the malicious Android apps render the conventional malware detection techniques ineffective,which

International Journal of Engineering Technology and Management Science
Website: ijetms.in Issue: 2 Volume No.9 March - April – 2025
DOI:10.46647/ijetms.2025.v09i02.093 ISSN: 2581-4621

results in a large number of malicious applications remaining undetected. This calls formore effective techniques for detection and classification of Android malware. Hence, in thispaper, present an Android malicious application detection framework based on the SupportVector Machine (SVM) and Active Learning technologies. In our approach, we extractapplications' activities while in execution and map them into a feature set, we then attachtimestamps to some features in the set and show that our novel use of time-dependent behaviortracking can significantly improve the malware detection accuracy. In particular, we build anactive learning model using Expected error reduction query strategy to integrate new informativeinstances of Android malware and retrain the model to be able to do adaptive online learning.The methods proposed in this related work contribute to key aspects and a higher predictive ratefor malware detection. Certain research has focused on increasing accuracy, while others havefocused on providing a larger dataset, some have been implemented by employing variousfeature sets, and many studies have combined all of these to improve detection rate efficiency. Inthe authors offer a system for detecting Android malware apps to aid in the organization of theAndroid Market. The proposed framework aims to provide a machine learning-based malwaredetection system for Android to detect malware apps and improve phone users' safety andprivacy. This system monitors different permission-based characteristics and events acquiredfrom Android apps and examines these features employing machine learning classifiers todetermine if the program is goodware or malicious.The authors present a unique Android malware detection approach dubbed Permission- basedMalware Detection Systems (PMDS) based on a study of samples of benign and maliciousAndroid applications. In PMDS, requested permissions are viewed as behavioral markers, and amachine learning model is built on those indicators to detect new potentially dangerousbehavior in unknown apps depending on the mix of rights they require. PMDS identifies morethan all heretofore unknown malware, with a false positive rate.The system is not implemented Machine Learning Algorithm And Ensemble Learning.➢ The system is not implemented Reverse Engineered Applications characteristics.

Implementation

1.Data Collection Module: The Data Collection Module serves as the foundation for the framework by gathering information from reverseengineered 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.

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 userfriendly front-end for system interaction. It displays relevant information on detected malware, allowing users to initiate

DOI:10.46647/ijetms.2025.v09i02.093 ISSN: 2581-4621

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.

Conclusion

In this research, devised a framework that can detect malicious Android applications. The proposed technique takes into account various elements of machine learning and achieves a 96.24% in identifying malicious Android applications. first define and pick functions to capture and analyze Android apps' behavior, leveraging reverse application engineering and AndroGuard to extract features into binary vectors and then use python build modules and split shuffle functions to train the model with benign and malicious datasets. Our experimental findings show that our suggested model has a false positive rate of 0.3 with 96% accuracy in the given environment with an enhanced and larger feature and sample sets. In the future work , also discovered that when dealing with classifications and high-dimensional data, ensemble and strong learner algorithms perform comparatively better.

References

[1] A. O. Christiana, B. A. Gyunka, and A. Noah, “Android Malware Detection throughMachine Learning Techniques: A Review,” Int. J. Online Biomed. Eng. IJOE, vol. 16, no. 02,p. 14, Feb. 2020, doi: 10.3991/ijoe.v16i02.11549. [2] D. Ghimire and J. Lee, “Geometric Feature-Based Facial Expression Recognition in ImageSequences Using Multi-Class AdaBoost and Support Vector Machines,” Sensors, vol. 13, no.6, pp. 7714–7734, Jun. 2013, doi: 10.3390/s130607714. [3] R. Wang, “AdaBoost for Feature Selection, Classification and Its Relation with SVM, AReview,” Phys. Procedia, vol. 25, pp. 800–807, 2012, doi: 10.1016/j.phpro.2012.03.160. [4] J. Sun, H. Fujita, P. Chen, and H. Li, “Dynamic financial distress prediction with conceptdrift based on time weighting combined with Adaboost support vector machine ensemble,”Knowl.-Based Syst., vol. 120, pp. 4–14, Mar. 2017, doi: 10.1016/j.knosys.2016.12.019. [5] A. Garg and K. Tai, “Comparison of statistical and machine learning methods inmodelling of data with multicollinearity,” Int. J. Model. Identif. Control, vol. 18, no. 4, p.295, 2013, doi: 10.1504/IJMIC.2013.053535. [6] C. P. Obite, N. P. Olewuezi, G. U. Ugwuanyim, and D. C. Bartholomew,“Multicollinearity Effect in Regression Analysis: A Feed Forward Artificial Neural NetworkApproach,” Asian J. Probab. Stat., pp. 22– 33, Jan. 2020, doi: 10.9734/ajpas/2020/v6i130151. [7] W. Wang et al., “Constructing Features for Detecting Android Malicious Applications:Issues, Taxonomy and Directions,” IEEE Access, vol. 7, pp. 67602–67631, 2019, doi:10.1109/ACCESS.2019.2918139.