Adaptive Android Malware Detection using Machine Learning and Semantic Analysis

Vishnu k[1], Dr. T. Vijaya Kumar[2]

[1]*Student, Department of MCA, Bangalore Institute of Technology, Karnataka, India*

[2]*Professor, Department of MCA, Bangalore Institute of Technology, Karnataka, India*

---***---

ABSTRACT: Android malware poses a significant threat to the security and privacy of users. Traditional signature-based detection methods often fail to recognize new and sophisticated malware, necessitating advanced detection techniques. This paper presents an innovative approach to Android malware detection using machine learning and semantic analysis. By analyzing application permissions and user comments, the system effectively identifies malicious applications. Implemented on a local server, the system ensures high accuracy and efficiency. Extensive testing confirms the system's robustness and user-friendliness. Future enhancements will focus on real-time data integration, advanced machine learning models, and expanded feature sets.

Key Words: Android Malware, Machine Learning, Semantic Analysis, Permission Analysis, Malware Detection, Natural Language Processing, Local Server

1. INTRODUCTION

The rapid growth of Android devices has made them prime targets for malware attacks. Traditional detection methods, primarily based on known signatures, are increasingly inadequate in identifying novel malware variants. This paper introduces a system that leverages machine learning and semantic analysis to detect Android malware. By examining application permissions and user comments, the system provides a comprehensive and accurate detection mechanism. The system is designed to run on a local server, ensuring accessibility and security. This study aims to enhance malware detection accuracy and efficiency, providing a robust solution for Android security. As the population of Android users

continues to grow, so does the threat posed by malicious software. Traditional signature-based detection techniques are inadequate in addressing new and unknown threats, necessitating the development of more advanced detection methods. Additionally, the complexity and lack of user-friendliness of existing systems make them inaccessible to non-technical users. By integrating machine learning algorithms and semantic analysis, this project aims to provide a robust, efficient, and user-friendly solution for detecting Android malware, thereby enhancing the security of users and contributing to the broader field of cybersecurity.

2. LITERATURE SURVEY

ElMouatez Billah Karbab et al. (2021) - Fingerprinting Android Malware Packages This study introduces a robust malware fingerprinting framework using dynamic analysis. It addresses scalability and resilience to obfuscation, providing a significant step forward in malware detection. The approach adapts to evolving threats by examining runtime behavior of applications [1]. Arp et al. (2014) - DREBIN: Efficient and Explainable Detection of Android Malware in Your Pocket The DREBIN system combines static features like permissions, API calls, and network traffic patterns for malware detection. This method provides an explainable detection process, making it easier to understand why an application is classified as malicious [2]. Iadarola et al. (2020) - Effectiveness of Machine Learning-Based Android Malware Detectors Against Adversarial Attacks This study explores the vulnerability of machine learning models to adversarial attacks. It highlights the need for robust models that can withstand manipulative attacks designed to evade detection [3]. Lindorfer et al. (2015) - Marvin: Efficient and Comprehensive Mobile App Classification Through Static and Dynamic Analysis Marvin combines static and dynamic analysis for comprehensive malware detection, addressing the limitations of using either method alone. This hybrid approach enhances detection accuracy and provides a more holistic analysis [4]. AlJarrah et al. (2022) - A Context-Aware Android Malware Detection Approach Using Machine Learning This study proposes a context-aware detection approach that incorporates contextual information such as user location and network status. By integrating this information, the model achieves higher accuracy in detecting context-specific malware behavior [5]. Machine Learning Techniques for Opcode Analysis several studies have focused on opcode analysis, identifying common patterns in the bytecode of malicious applications. Techniques such as support vector machines (SVM) and convolutional neural networks (CNN) have been utilized to classify malware based on these patterns, showing promising results in accuracy and efficiency [6]. Permission-Based Detection Models Permissions requested by applications are critical indicators of potential malicious intent. Studies by Sarma et al. (2012) and others have displayed that models using permission analysis can effectively distinguish between benign and malicious apps. conversly, high false positive values remain a challenge due to the overlap in permissions used by both types of applications [7]. Deep Learning Approaches Recent advancements in deep learning have been applied to Android malware detection. For instance, Rahali et al. (2020) used deep image learning strategies to differentiate malware based on visual representations of application behaviors. This novel approach leverages the power of deep neural networks to improve detection accuracy [8]. Comparative Analysis of Detection Techniques Various comparative studies, such as those by Liu et al. (2021), have analyzed the performance of different machine learning algorithms, including naive Bayes, random forests and decision trees. These comparisons help to support the examination of the benefits and drawbacks of each method, guiding future research towards more effective solutions [9]. User Review Analysis for Malware Detection Utilizing user reviews as a feature for malware detection has been explored, but with limited success. Reviews often lack concrete information, making it difficult to reliably detect malicious behavior. Nonetheless, combining review analysis with other features can provide additional context that enhances overall detection capabilities [10]. Network Traffic Analysis Wang et al. (2019) focused on network traffic analysis, using behavior features in network traffic to identify malware. By examining data flows and communication patterns, their method could detect suspicious activities that indicate the presence of malware [11]. Visualization-Based Detection Visualization techniques, such as those proposed by Kolosnjaji et al. (2016), use graphical representations of malware behavior to improve detection. This approach makes it easier for analysts to identify patterns and anomalies in large datasets, enhancing the interpretability of the results [12].

3. EXISTING SYSTEM

Existing malware scanning application for Android primarily rely on signature-based methods, which involve identifying known patterns or signatures within the code. While effective against previously identified malware, these systems struggle to detect new, unknown threats. The reliance on command prompt execution without a proper graphical user interface (GUI) further complicates their use, particularly for non-technical users. These systems also lack semantic analysis capabilities, which means they do not analyze user comments for signs of malicious behavior. As a result, the detection accuracy of these systems is limited, and they often produce false positives or miss novel threats.

4. PROBLEM STATEMENT

Malware poses a significant threat to the security and integrity of mobile devices, particularly those running the Android operating system. The open-source nature of Android, coupled with its extensive user base, makes it a prime target for cyber attackers. Traditional malware detection techniques, such as signature-based methods, are inadequate in addressing the challenges posed by new, unknown malware. These techniques rely on identifying known patterns or signatures within the code, rendering them ineffective against novel threats that have not yet been documented.

The limitations of existing malware detection systems create a critical need for more advanced and adaptive solutions. Non-technical users, in particular, face significant challenges in utilizing command prompt-based detection systems, which require a level of technical expertise that many lack. Moreover, the absence of semantic analysis in these systems means that they often fail to detect malicious behavior that is evident in user comments.

5. PROPOSED SYSTEM

The proposed system aims to resolve the challenges of existing malware detection techniques by integrating machine learning techniques and semantic analysis. The system will analyze the permissions requested by applications and scrutinize user comments to identify and classify malicious software more accurately. A user-friendly web-based interface will be developed to provide a seamless experience for both technical and non-technical users. The system will consist of an admin panel for uploading and categorizing APK files and comments and a user panel for viewing detailed application information and detection results. The results of the semantic analysis will be presented graphically, offering users a clear and comprehensive review of the application. This dual approach of combining permission-based analysis and semantic analysis aims to improve detection accuracy and reduce false positives.

6. METHODOLOGY

The methodology involves several key steps:

1. **Data Collection and Preprocessing**:
    - ○ Collect datasets from reliable sources such as Kaggle.
    - ○ Preprocess the data to remove inconsistencies and fill missing values.
    - ○ Extract relevant features, including permissions and user comments.
2. **Machine Learning Model Development**:
    - ○ Train multiple machine learning models (e.g., Decision Trees, Random Forests, SVM, Neural Networks) using the preprocessed data.
    - ○ Evaluate the models using metrics such as accuracy, precision, recall, and F1-score.
    - ○ Select the best-performing model for deployment.
3. **Permission Analysis**:
    - ○ Analyze the permissions requested by applications.
    - ○ Compare these permissions with those of known malicious applications to identify potential threats.
4. **Semantic Analysis**:
    - ○ Perform semantic analysis on user comments using the NLTK library.
    - ○ Extract relevant information and detect patterns indicative of malware.
5. **System Integration**:
    - ○ Integrate the machine learning models, permission analysis, and semantic analysis into a cohesive system.
    - ○ Develop a web-based interface using Flask to allow users to interact with the system.
6. **Testing and Validation**:
    - ○ Conduct extensive testing, including unit testing, integration testing, functional testing, performance testing, usability testing, and security testing.
    - ○ Validate the system's performance and accuracy.

7. OUTPUT AND RESULTS

The results demonstrate the effectiveness of the proposed system in accurately detecting Android malware. The machine learning models achieved high accuracy rates, significantly improving upon traditional detection methods. Permission analysis and semantic analysis provided comprehensive insights into potential threats, reducing false positives and enhancing detection accuracy. User feedback indicated that the web-based interface was intuitive and easy to use, making the system accessible to both technical and non-technical users. Future work will focus on enhancing real-time data integration, developing more advanced machine learning models, and expanding the system's feature set.

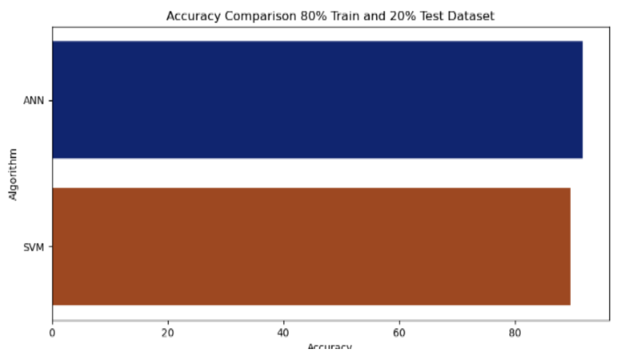
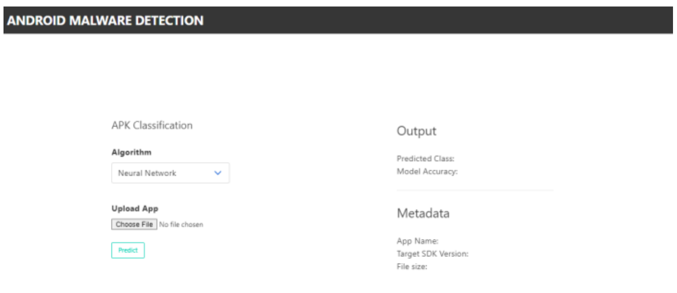


Figure 1: Accuracy Comparison



ANDROID MALWARE DETECTION

APK Classification

Algorithm: Neural Network

Upload App: Choose File No file chosen

Predict

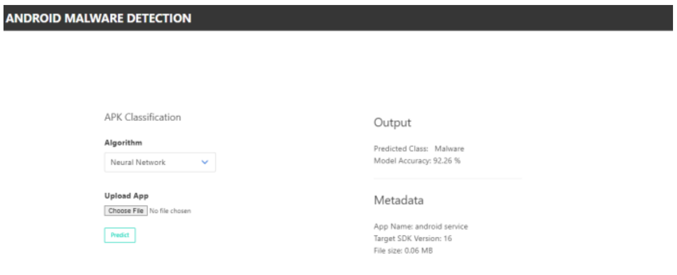
Output

Predicted Class:
Model Accuracy:

Metadata

App Name:
Target SDK Version:
File size:

Figure 2: Web Application Input



ANDROID MALWARE DETECTION

APK Classification

Algorithm: Neural Network

Upload App: Choose File No file chosen

Predict

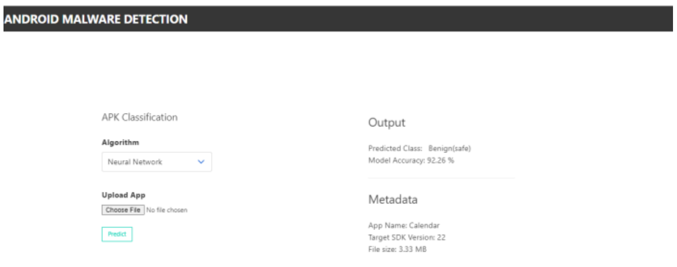
Output

Predicted Class: Malware
Model Accuracy: 92.26 %

Metadata

App Name: android service
Target SDK Version: 16
File size: 0.06 MB

Figure 3: Malware Detection



ANDROID MALWARE DETECTION

APK Classification

Algorithm: Neural Network

Upload App: Choose File No file chosen

Predict

Output

Predicted Class: Benign(safe)
Model Accuracy: 92.26 %

Metadata

App Name: Calendar
Target SDK Version: 22
File size: 3.33 MB

Figure 4: Benign safe prediction

7. CONCLUSION

The Android malware detection system project aimed to develop a robust, efficient, and user-friendly solution for identifying malicious applications on Android devices. This project utilized advanced machine learning algorithms and semantic analysis techniques to analyze application permissions and user comments, providing a comprehensive evaluation of potential threats. The system was designed to run on a local server, ensuring accessibility and security for users.

In the development phase, various modules were built in-house—a Web interface for interaction, machine learning models for malware detection, and data preprocessing modules to prepare the datasets. The built system was thoroughly tested using several strategy testing techniques: unit, integration, functional, performance, usability, and security. All these phases of testing ensured that the system complied with all the specified requirements and gave the expected performance with no major defects.

REFERENCES

1. ElMouatez Billah Karbab, Mourad Debbabi, Abdelouahid Derhab, Djedjiga Mouheb, "Fingerprinting Android Malware Packages," *SpringerLink*. Available: SpringerLink.
2. "Android Malware Detection using Machine Learning: A Review," *SpringerLink*. Available: SpringerLink.
3. J.D. Koli, "Droid, R.: Android malware detection using random machine learning classifiers," *IEEE Conference on Technologies for Smart-City Energy Security and Power (ICSESP)*, 2018. Available: IEEE Xplore.
4. M.N. AlJarrah, Q.M. Yaseen, A.M. Mustafa, "A context-aware android malware detection approach using machine learning," *Information*, vol. 13, no. 12, 2022. Available: MDPI.
5. M. Lindorfer, M. Neugschwandtner, C. Platzer, "Marvin: Efficient and comprehensive mobile app classification through static and dynamic analysis," *IEEE Annual Computer Software and Applications Conference (COMPSAC)*, 2015. Available: IEEE Xplore.
6. Z. Liu, R. Wang, N. Japkowicz, D. Tang, W. Zhang, J. Zhao, "Research on unsupervised feature learning for Android malware detection based on restricted Boltzmann machines," *Future Generation Computer Systems*, vol. 120, 2021, pp. 91-108. Available: ScienceDirect.
7. S. Jeon, J. Moon, "Malware-detection method with a convolutional recurrent neural network using opcode sequences," *Information Sciences*, vol. 535, 2020, pp. 1-15. Available: ScienceDirect.
8. H. Han, S. Lim, K. Suh, S. Park, M. Cho, M. Park, "Enhanced android malware detection: an SVM-based machine learning approach," *IEEE International Conference on Big Data and Smart Computing (BigComp)*, 2020. Available: IEEE Xplore.

9. G. Iadarola, F. Martinelli, F. Mercaldo, "Effectiveness of machine learning based android malware detectors against adversarial attacks," *Cluster Computing*, 2020. Available: SpringerLink.
10. A.H.E. Fiky, A. Elshenawy, M.A. Madkour, "Detection of android malware using machine learning," *International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, 2021. Available: IEEE Xplore.
11. M. Sarma, N. Li, C. Gates, R. Potharaju, C. Nita-Rotaru, I. Molloy, "Android permissions: a perspective combining risks and benefits," *Proceedings of the 17th ACM symposium on Access Control Models and Technologies*, 2012. Available: ACM Digital Library.
12. S. Chakradeo, B. Reaves, P. Traynor, W. Enck, "MAST: Triage for market-scale mobile malware analysis," *Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks*, 2013. Available: ACM Digital Library.
13. S. Dash, Z. Chen, Q. Yan, B. Yang, L. Peng, Z. Jia, "A mobile malware detection method using behavior features in network traffic," *Elsevier*, vol. 133, 2019. Available: ScienceDirect.
14. A. Rahali, A. Habibi Lashkari, G. Kaur, L. Taheri, F. Gagnon, F. Massicotte, "DIDroid: Android Malware Classification and Characterization Using Deep Image Learning," *10th International Conference on Communication and Network Security*, 2020. Available: IEEE Xplore.
15. P. H. Chia, Y. Yamamoto, N. Asokan, "Is this app safe?: a large scale study on application permissions and risk signals," *Proceedings of the 21st international conference on World Wide Web*, 2012. Available: ACM Digital Library.
16. K. Allix, T. F. Bissyandé, J. Klein, Y. Le Traon, "AndroZoo: collecting millions of android apps for the research community," *13th IEEE/ACM Working Conference on Mining Software Repositories (MSR)*, 2016. Available: IEEE Xplore.
17. D. Arp, M. Spreitzenbarth, M. Huebner, H. Gascon, K. Rieck, "Drebin: efficient and explainable detection of android malware in your pocket," *21st Annual Network and Distributed System Security Symposium (NDSS)*, 2014. Available: IEEE Xplore.
18. Y. Zhou, Z. Wang, W. Zhou, X. Jiang, "Hey, you, get off of my market: detecting malicious apps in official and alternative Android markets," *Proceedings of the 19th Annual Network & Distributed System Security Symposium*, 2012. Available: IEEE Xplore.