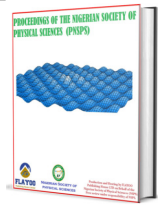
Published by Nigerian Society of Physical Sciences. Hosted by FLAYOO Publishing House LTD

Proceedings of the Nigerian Society of Physical Sciences

Journal Homepage: https://flayoophl.com/journals/index.php/pnspsc



Android malware detection using random forest algorithm

Samson Isaac[a,*], Abdullahi Tanimu[a], Mohammed Shamsuddeen Tukur[a], Jacob Isaac[b], Amina Bala Ja'afaru[a]

[a]*Department of Computer Science, Kaduna State University, Kaduna Nigeria*
[b]*Department of Agriculture, Kaduna State College of Education, Gidan Waya, Kaduna Nigeria*

A B S T R A C T

The proliferation of mobile devices and their dependence on the Android OS has made them prime targets for cybercriminals, leading to an escalating threat of malware. This study addresses the growing need for effective malware detection methods by exploring the application of machine learning (ML) techniques to enhance the security of Android devices. Specifically, the research investigates the performance of various ML algorithms, with a focus on Random Forest, in detecting malware on the Android platform. In this study, a subset of the Android dataset, which consists of: 50,000 benign Android applications and 50,000 malicious Android applications. Through comprehensive analysis and experimentation, the study demonstrates significant improvements in detection accuracy achieving 0.99 accuracy. Other key performance metrics, including accuracy, recall, precision, and F1-Score. These results highlight the potential of ML to revolutionize Android OS malware detection, offering robust, real-time protection against evolving threats while minimizing the impact on device performance. The findings contribute valuable insights for cybersecurity practitioners, mobile app developers, and researchers, paving the way for more secure mobile environments and advanced malware detection systems.

Keywords: Android, Random forest, Malware, Mobile device.

DOI:10.61298/pnspsc.2025.2.178

1. INTRODUCTION

In today's digital age, the rapid proliferation of mobile devices has led to an increased reliance on the Android operating system (OS), making Android devices a prime target for cybercriminals. As the popularity of these devices continues to soar, so does the sophistication of malicious actors seeking to exploit vulnerabilities within the OS. This surge in mobile threats puts the security and privacy of millions of users at significant risk. Cybercriminals constantly innovate, developing new and more complex techniques to breach Android OS defenses, leading to a growing threat landscape [1].

Traditional malware detection methods, such as signature-based techniques, are becoming increasingly inadequate in the face of these evolving threats. To address this challenge, researchers have turned to machine learning (ML), a branch of artificial intelligence (AI) that focuses on teaching computers to recognize patterns and make informed decisions based on data. By training ML algorithms on vast datasets of known malware and benign programs, researchers can identify unique patterns and features that distinguish malicious software from legitimate applications, thus enhancing the detection capabilities of security

*Corresponding Author Tel. No.: +234-806-3232-169.
e-mail: abdullahi.tanimu@kasu.edu.ng (Samson Isaac)

systems [2]. The potential of ML in Android OS malware detection lies in its ability to adapt to the constantly changing nature of malware. ML-based systems offer real-time detection, flexibility, scalability, and a higher degree of accuracy compared to traditional methods. However, the effectiveness of these systems depends on overcoming several critical challenges. One of the primary obstacles is the acquisition of diverse and representative datasets that include both malicious and benign applications. Privacy concerns and the need for continuous updates to keep pace with emerging threats make data collection a complex and ongoing task [3].

Another challenge is the selection of the most relevant features from large datasets. Identifying the right features is crucial, as irrelevant or redundant information can negatively impact the performance of the algorithm. Researchers must also rigorously evaluate various ML algorithms to determine the most suitable for Android OS malware detection, while continuously refining them to minimize false positives and negatives [4].

The future of Android malware detection lies in harnessing the full potential of ML. By addressing the challenges of data gathering, feature selection, and algorithm optimization, researchers can develop more robust and efficient malware detection systems. This research aims to contribute to this evolving field by exploring advanced ML techniques, including supervised, unsupervised, and deep learning methods, to differentiate between malicious and legitimate Android applications. The ultimate goal is to enhance the security and privacy of Android users while providing valuable insights and tools for cybersecurity practitioners, mobile app developers, and policymakers [5]. By focusing on these cutting-edge strategies, this project seeks to play a pivotal role in combating the ever-growing threat of Android malware, ensuring the safety and integrity of mobile devices worldwide.

2. RELATED WORKS

Powering billions of smartphones, tablets, and other devices throughout the globe, Android OS has emerged as the leading OS in the mobile device industry. Its open-source status and extensive ecosystem of compatible apps are major factors in its widespread use. But there are major security concerns that come along with its broad [6]. Protecting user data and device integrity, this section delves into the architecture of the Android OS and its strong security measures. People, businesses, and the whole digital ecosystem are all at risk from malware. In order to create effective protection tactics and increase awareness about cybersecurity, it is essential to understand typical vulnerabilities that malware exploits. Inadequate security measures, such as using a weak password or falling victim to a phishing attempt, are common problems [7, 8]. Regular software patches may protect against vulnerabilities in outdated versions, but phishing attempts use false email, bogus websites, or instant messaging to trick their targets. It is crucial to regularly update and patch systems since unpatched systems may exploit known vulnerabilities. Password cracking tools like brute-force and dictionary assaults may easily break through weak passwords, giving hackers access to user accounts and sensitive data [9]. Pretexting, baiting, tailgating, and scareware are all examples of social engineering techniques that take advantage of people's emotions and rationality to get beyond security protocols. Malware vulnerabilities may be reduced by routine software upgrades, education of users, strong password habits, security of email and websites, and security audits. People and businesses may greatly lessen their vulnerability to malware attacks by maintaining up-to-date software, encouraging user education, using secure password habits, and routinely assessing security measures [3, 10]. The absence of adversarial resilience is the most significant weakness, as it allows malware developers to evade detection systems that rely on ML. The detection accuracy and reliability in real-world circumstances may be improved by strengthening models against these assaults. This should be the focus of future study. There is also a lack of study on Android malware's dynamic analysis and runtime detection. Determining zero-day vulnerabilities and polymorphic malware relies heavily on dynamic behaviors, system calls, API interactions, and network data. Researchers should look at dynamic analytic methods using ML for real-time malware detection and threat adaptation in future research [11].

Many studies that were conducted recently failed to use a user-centric and context-aware approach. If we take into account the user's choices, habits, app use, and context, we may greatly improve detection accuracy and reduce false positives. Improved detection performance might be achieved by the development of ML models that combine user-centric attributes and contextual awareness in future research. The proliferation of IoT devices has left us in the dark about how malware might spread over networks that include Android and other mobile platforms [12]. If we want to build effective detection and prevention methods, researchers need to look at the security consequences of interactions, vulnerabilities, and the routes that malware uses to spread among IoT devices.

Since ML-based malware detection systems need access to private data and device information, privacy considerations are of utmost importance. The security of user data should not be compromised in any way, and future research should concentrate on ML approaches that preserve privacy, such as encrypted model inference, differential privacy, and federated learning [13]. Ultimately, strengthening cybersecurity defenses,

3. METHODOLOGY

The primary sources of data consist of public malware repositories like as VirusTotal, MalwareBazaar, and AndroZoo. The study uses two Android dataset, which consists of: 50,000 benign Android applications and 50,000 malicious Android applications. The malicious applications were labeled as such based on the VirusTotal reports, which aggregate the results of multiple antivirus scanners. Feature Extraction to represent the Android applications, extracted the following set of features: These repositories provide access to extensive datasets of Android malware samples that have been gathered from a variety of sources. For training and testing ML models, research datasets like the Genome dataset, the AMD dataset, and the Drebin dataset provide labeled examples of malware and benign applications. The dynamic analysis of malware samples is provided by platforms such as Cuckoo Sandbox, FireEye, and McAfee. These systems capture runtime behaviors, API calls, network traffic, and system interactions; they also enable dynamic analysis. For comparative research and feature extraction, app stores and repositories

Table 1. Comparison with existing studies.

Study	Algorithm	Accuracy	Recall	F1-Score	Precision
Manzil and Manohar (2023)	Random Forest	0.931	0.931	0.931	0.931
Odat and Yaseen (2023)	Random Forest	0.95	-	-	-
This Study	Random Forest	0.99	0.99	0.99	0.99

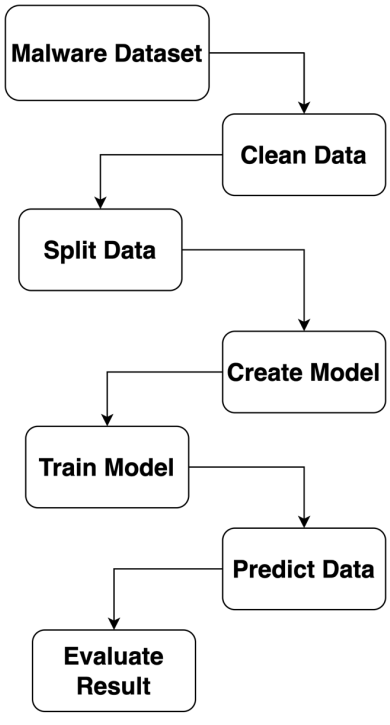


Figure 1. Standard approach for detecting malware.

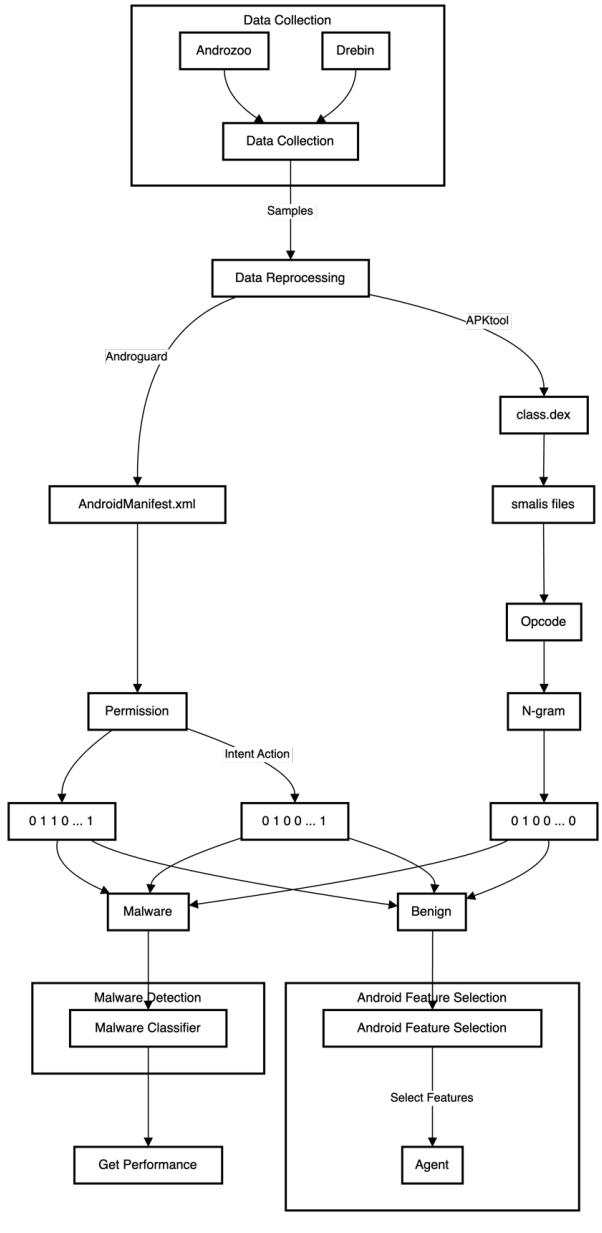


Figure 2. Proposed approach for detecting malware.

let users have access to a huge number of safe applications [14].

Extraction of features, normalization, and scaling of the data, feature selection, management of missing values, and data balance are all procedures that are included in the preparation of the data. To represent each sample in a meaningful feature space, feature extraction is the process of extracting important features from raw data [15]. Permissions, API calls, manifest properties, code analysis, normalization and scaling, managing missing values, feature selection, and data balance are all examples of features that are often used. Numerical features are subjected to normalization techniques such as Min-Max scaling or Z-score normalization to bring them within a common scale or range. This ensures that features with different scales or units contribute equally to the model during training and eliminates biases that are caused by the magnitudes of the features [16]. To minimize the dimensionality of the data and pick the features that are most relevant for model training, the Recursive Feature Elimination (RFE) are used [17]. Learning and predicting models may be affected by how datasets handle missing values. The right handling of missing data may be accomplished by the use of several strategies, including imputation, the deletion of missing data points, or the utilization of robust algorithms that are resilient to missing values. It is vital to implement data balance to handle developing vulnerabilities in the Android OS's security. Several methods, including oversampling minority classes, undersampling majority classes, and the use of synthetic data generation (SMOTE), are utilized to achieve the goal of effectively balancing class distributions and enhancing the performance of models [18, 19].

The detection of malware on Android with ML relies heavily on data gathering and preprocessing. Researchers and practitioners of cybersecurity can construct efficient ML models for the detection of malware and boost overall cybersecurity defenses in the Android ecosystem by utilizing a wide variety of

data sources, extracting features that are pertinent to the problem at hand, and applying preprocessing steps such as normalization, feature selection, and data balancing. When it comes to improving research and tackling new risks in Android OS security, collaborative efforts, standardized datasets, and creative preprocessing approaches are all critical components [20]. The existing standard approach is shown in Figure 1 and the proposed approach for detecting malware is shown in Figure 2.

4. RESULT AND DISCUSSIONS

Table 1 summarizes the performance of three related studies of machine learning models on the test dataset for malware detection in Android OS. The table 1 shows the performance of Random Forest algorithms in Android OS malware detection across three studies, highlighting key metrics such as accuracy, recall, F1-Score, and precision. The current study outperforms the others, achieving near-perfect scores of 0.99 across all metrics, indicating a highly accurate and reliable model. Ref. [21] reported a consistent performance with all metrics at 0.931, demonstrating balanced effectiveness in malware detection. Meanwhile, Ref. [22] achieve a slightly higher accuracy of 0.95 but lack reported values for recall, F1-Score, and precision, limiting a full evaluation of their model's performance. Overall, the progression in these studies reflects significant improvements in the optimization and effectiveness of Random Forest models, with the current study setting a new benchmark for accuracy and reliability in Android malware detection.

The results of the comparative analysis of the different machine learning models for Android malware detection discussion are as follows.

Random forest (RF): The proposed RF model achieved the best overall performance, with the highest accuracy (0.99), precision (0.99), recall (0.99) and F1-score (0.99). The RF's ability to automatically learn relevant features from the raw input data, combined with its effective feature extraction and classification capabilities, made it the top-performing model in this study.

5. CONCLUSION

This study has demonstrated the significant potential of machine learning (ML) techniques, particularly the Random Forest algorithm, in enhancing the detection of malware on Android OS. By achieving near-perfect performance metrics 0.99 in accuracy, recall, precision, and F1-Score. this research highlights the effectiveness of ML in providing robust, real-time protection against the evolving threat of malware. The findings suggest that ML-based models can significantly improve the security of Android devices, offering a powerful tool for safeguarding user data and ensuring system integrity. Despite the success of the Random Forest algorithm in this study, several challenges remain, such as optimizing feature selection, addressing data imbalance, and developing lightweight models that do not compromise device performance. These challenges point to the need for ongoing research to refine ML techniques further and ensure their practical implementation in real-world scenarios. Additionally, the study emphasizes the importance of interdisciplinary collaboration and standardized evaluation criteria to advance the field of Android OS malware detection. This research not only sets a new benchmark for the performance of malware detection systems but also lays the groundwork for future advancements in mobile cybersecurity. As cyber threats continue to evolve, the development and implementation of sophisticated ML models will be crucial in maintaining the security and privacy of Android users worldwide.

DATA AVAILABILITY

The data will be available on request from the corresponding author.

References

[1] M. Odusami, O. Abayomi-Alli, S. Misra, O. Shobayo, R. Damasevicius & R. Maskeliunas, *Android malware detection: a survey*, in *Applied Informatics*, Communications in Computer and Information Science, Springer International Publishing, 2018. https://doi.org/10.1007/978-3-030-01535-0_19.

[2] A. B. Riad, "Plugin-based tool for teaching secure mobile application development", Information Systems Education Journal **19** (2022) 25. https://files.eric.ed.gov/fulltext/EJ1297704.pdf.

[3] E. J. AAlqahtani, R. Zagrouba & A. Almuhaideb, "A survey on android malware detection techniques using machine learning Algorithms", 6th International Conference on Software Defined Systems, (SDS), 2019, pp. 110-117. https://doi.org/10.1109/SDS.2019.8768729.

[4] W. T. Sung, "Smart garbage bin based on AIoT", Intelligent automation and soft computing **32** (2022) 1387. https://doi.org/10.32604/IASC.2022.022828.

[5] S. Modgil, "AI technologies and their impact on supply chain resilience during covid-19", International Journal of Physical Distribution and Logistics Management **52** (2022) 130. https://doi.org/10.1108/IJPDLM-12-2020-0434.

[6] F. Xu, "Android on PC: on the security of end-user android emulators", Proceedings of the ACM Conference on Computer and Communications Security, 2021, pp. 1566. https://doi.org/10.1145/3460120.3484774.

[7] F. A. Alaba, "Ransomware attacks on remote learning systems in 21st century: a survey", Biomedical Journal of Scientific & Technical Research **35** (2021) 27322. https://doi.org/10.26717/bjstr.2021.35.005649.

[8] S. Isaac, D. K. Ayodeji, Y. Luqman, S. M. Karma & J. Aminu, "Cyber security attack detection model using semi-supervised learning", FUDMA Journal of Sciences **8** (2024) 92.

[9] M. Humayun, "Internet of things and ransomware: evolution, mitigation and prevention", Egyptian Informatics Journal **22** (2021) 105. https://doi.org/10.1016/j.eij.2020.05.003.

[10] D. Su, "Detecting android locker-ransomware on chinese social networks", IEEE Access **7** (2019) 20381. https://doi.org/10.1109/ACCESS.2018.2888568.

[11] N. Saleem, "Enhancing security of android operating system based phones using quantum key distribution", EAI Endorsed Transactions on Scalable Information Systems **7** (2020) 28. https://doi.org/10.4108/eai.13-7-2018.165281.

[12] J. Rathod & D. Bhatti, *Minimization of attributes for the detection of vulnerabilities in android applications*, New Frontiers in Communication and Intelligent Systems, SCRS, India, 2022, pp. 475-487. https://doi.org/10.52458/978-81-95502-00-4-49.

[13] X. Zhan, "ATVHunter: reliable version detection of third-party libraries for vulnerability identification in android applications", 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), Madrid, ES, 2021, pp. 1695-1707. https://doi.org/10.1109/ICSE43902.2021.00150.

[14] A. Stroumpoulis & E. Kopanaki, "Theoretical perspectives on sustainable supply chain management and digital transformation: a literature review and a conceptual framework", Sustainability (Switzerland) **14** (2022) 8. https://doi.org/10.3390/su14084862.

[15] M. Kehayov, L. Holder & V. Koch, "Application of artificial intelligence technology in the manufacturing process and purchasing and supply management", Procedia Computer Science **200** (2019) 1209. https://doi.org/10.1016/j.procs.2022.01.321.

[16] M. Regona, "Opportunities and adoption challenges of AI in the construction industry: a PRISMA review", Journal of Open Innovation: Technology, Market, and Complexity **8** (2022) 45. https://doi.org/10.3390/joitmc8010045.

[17] A. Ghosh, D. J. Edwards & M. R. Hosseini, "Patterns and trends in internet

of things (IoT) research: future applications in the construction industry'', Engineering, Construction and Architectural Management **28** (2021) 457. https://doi.org/10.1108/ECAM-04-2020-0271.

[18] N. M. Kumar, ''Distributed energy resources and the application of an iot, and blockchain in smart grids'', Energies **13** (2020) 21. https://doi.org/10.3390/en13215739.

[19] S. Isaac, T. Abdullahi, M. S. Tukur, J. Isaac & A. B. Ja'afaru, ''Social media fake news detection model using support vector machine'', BIMA Journal of Science and Technology **9** (2025) 40. https://doi.org/10.56892/bima.v9i1A.1233

[20] M. Tavana, ''A review of digital transformation on supply chain process management using text mining'', Processes **10** (2022) 842. https://doi.org/10.3390/pr10050842.

[21] H. H. R. Manzil & S. Manohar Naik, ''Android malware category detection using a novel feature vector-based machine learning model'', Cybersecurity **6** (2023) 6. https://doi.org/10.1186/s42400-023-00139-.

[22] E. Odat & Q. M. Yaseen, ''A novel machine learning approach for android malware detection based on the co-existence of features'', IEEE Access **11** (2023) 15471. https://doi.org/10.1109/ACCESS.2023.3244656.