International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 03, March-2025

Advanced Malware Detection Frame Work using Random Forest

Amina A
Department of Computer Science
College of Engineering Karunagappally
Kerala, India

Aswathi M
Department of Computer Science
College of Engineering Karunagappally
Kerala, India

Abhirami B
Department of Computer Science
College of Engineering Karunagappally
Kerala, India

Amjad Ali N
Department of Computer Science
College of Engineering Karunagappally
Kerala, India

Dr Smitha Dharan Y
Department of Computer Science and Engineering
College of Engineering Karunagappally
Kerala, India

***Abstract*—With the rapid growth of mobile applications, ensuring security in digital marketplaces like Google Play Store has become a major challenge. Malicious apps can disguise themselves as legitimate software, leading to data breaches, financial fraud, and unauthorized access. This project presents an Advanced Malware Detection Framework using Random Forest (RF) to efficiently detect and classify harmful applications before they reach users. RF, a powerful ensemble learning algorithm, analyzes app features such as permissions, API calls, network activity, and behavioral patterns to distinguish between benign and malicious applications. By integrating this framework into app stores, security can be significantly enhanced, ensuring that users download only safe applications. Future enhancements may focus on real-time monitoring, feature refinement, and cloud-based integration for large-scale deployment, making this approach a scalable, efficient, and robust solution for malware detection in app stores.**

***Keywords*—Malware Detection, Random Forest, Cybersecu- rity, App Store Security, Google Play Store, Malicious Applica- tions, Ensemble Learning, API Calls Analysis, Network Activity Monitoring, Behavioral Analysis, Zero-Day Threats, Real-Time Detection, Feature Selection, Cloud-Based Security, Scalable Threat Detection.**

I. INTRODUCTION

The rapid evolution of malware has made traditional detection methods ineffective, particularly against sophisticated threats like polymorphic and zero-day malware. Signature-based approaches often fail to detect new or modified malware variants due to their reliance on predefined signatures. This project proposes an advanced malware detection framework using Random Forest (RF) to efficiently classify and detect malicious applications. RF, a robust machine learning algorithm, enhances accuracy by utilizing multiple decision trees, making it a scalable and adaptive solution for modern cybersecurity challenges. By integrating RF into malware detection systems, this framework aims to improve threat identification and minimize the limitations of traditional methods

II. PROBLEM STATEMENT

Traditional malware detection techniques, such as signature-based and static analysis methods, struggle to keep up with evolving threats, leading to high false-positive rates and poor adaptability. These systems lack scalability for large datasets and real-time detection, making them ineffective in real-world scenarios. The proposed framework addresses these issues by employing Random Forest, which analyzes multiple app features like permissions, API calls, and network activity to detect malware with greater accuracy. By leveraging RF's ability to handle large-scale data and identify complex patterns, this approach significantly enhances malware detection while reducing false positives.

III. OBJECTIVES

The primary objective of this project is to develop a high-accuracy malware detection system using Random Forest, ensuring effective classification of new and unknown threats. The system aims to enhance malware detection by leveraging RF's capability to process large datasets efficiently while maintaining low false-positive rates. It seeks to improve real-time detection, enabling security solutions for app stores like Google Play Store to prevent users from downloading malicious applications. Additionally, this project focuses on optimizing computational efficiency, ensuring scalability for large-scale cybersecurity applications.

IV. RANDOM FOREST

Random Forest (RF) is a powerful ensemble machine learning algorithm widely used for classification and regression tasks. It constructs multiple decision trees and combines their outputs through majority voting (for classification) or averaging (for regression), improving accuracy and reducing overfitting. RF is highly effective in handling large datasets, high-dimensional features, and complex patterns, making it an ideal choice for malware detection. In this project, RF is used to analyze various features of applications, such as permissions, API calls, network activity, and behavioral patterns, to classify them as benign or malicious. Its ability to handle imbalanced datasets, minimize false positives, and adapt to new and evolving threats makes it a robust solution for detecting malware in app stores like Google Play Store. By leveraging RF's efficiency and scalability, the proposed framework ensures real-time malware detection, enhancing overall cybersecurity in digital marketplaces.

V. CONTRIBUTIONS

This project contributes to malware detection by implementing Random Forest to improve accuracy, efficiency, and adaptability. Key contributions include feature-based malware classification, allowing for the detection of zero-day threats, and optimization of feature selection to enhance detection speed. The system supports real-time scanning, ensuring app store security by preventing malware distribution. Additionally, the framework minimizes false positives while maintaining high detection rates, making it a scalable and robust solution for cybersecurity applications.

VI. LITERATURE REVIEW

The challenge of malware detection has grown significantly due to the rapid emergence of new and unknown threats that often evade traditional detection methods. Conventional signature-based approaches struggle to keep up with evolving malware techniques, making them ineffective against polymorphic and zero-day attacks. To address these limitations, machine learning-based approaches, particularly those using Random Forest (RF), have gained prominence in cybersecurity research. RF is known for its ability to process large datasets efficiently while maintaining high accuracy and low false-positive rates. Its ensemble nature, which involves constructing multiple decision trees and aggregating their outputs, enhances model robustness and adaptability in detecting sophisticated malware.

Several studies have explored the application of Random Forest in malware detection, highlighting its capability to analyze behavioral patterns, API calls, permissions, and network activity to classify applications as benign or malicious. Unlike traditional methods, RF does not rely on predefined signatures but instead learns from historical data to detect new threats. Its ability to handle imbalanced datasets, noisy data, and high-dimensional feature spaces makes it a strong candidate for large-scale malware classification. Moreover, RF has demonstrated effectiveness in real-time app store security, where rapid and accurate threat detection is crucial for preventing malicious applications from reaching users.

One of the key advantages of RF is its ability to scale efficiently while minimizing computational overhead. Research has shown that RF maintains high detection accuracy while being computationally less expensive compared to deep learning models like CNNs. Additionally, its resistance to overfitting and ease of interpretability make it a practical choice for real-world cybersecurity applications. However, some studies indicate that RF may have limitations in detecting highly obfuscated malware, necessitating continuous model updates and feature optimization.

Overall, Random Forest has proven to be a reliable and efficient solution for malware detection, offering a balance between performance, scalability, and resource efficiency. Its application in app store security, particularly in detecting malicious applications before distribution, makes it an essential tool in modern cybersecurity frameworks. Future research can focus on enhancing RF's adaptability through incremental learning and integrating real-time detection mechanisms to further improve malware defense systems.

Song-Kyoo Kim et al. [1] explored various machine learning approaches like Random Forest (RF), Gradient Boosted Decision Trees (GBDT), and Support Vector Machines (SVM) using the CDL dataset. These models achieved high accuracy (95.17

S. A. Roseline et al. [2] proposed the Deep Random Forest (DRF) method for malware detection by converting malware binaries into grayscale images. They evaluated the model on datasets such as Malimg, BIG 2015, and MaleVis, achieving high accuracy rates of up to 98.65%, which surpassed traditional deep learning models. The DRF method demonstrated efficiency with lower complexity compared to other models, making it a practical choice for malware detection tasks. However, it faced challenges with the misclassification of similar malware types, such as Obfuscator.ACY and Ramnit, highlighting areas for further improvement.

A. A. Al-Hashmi et al. [3] proposed the Multifaceted Deep Ensemble Behavioral Malware Variant Detection Scheme (MDEB-MVDS-XGB), which integrates deep learning with Extreme Gradient Boosting (XGB) to analyze malware behavior based on API calls, network traffic, and file access patterns. This model achieved an impressive accuracy of 99.23%, demonstrating exceptional performance in detecting evasive malware. However, it is computationally intensive and slightly less effective when handling highly evasive malware variants, indicating potential areas for enhancement.

Jeon, J., Park, J. H., Jeong, Y. S. [4] proposed a Convolutional Neural Network (CNN) approach for IoT malware detection by transforming behavioral data into images. This method achieved a high accuracy of 99.28% and maintained a low false positive rate of 0.63%, making it highly effective in detecting both known and variant malware. However, it demands substantial computational power and is susceptible to evasion tactics if malware identifies the virtual environment used for analysis.

Zhang, X., Wang, J., Xu, J., Gu, C. [5] proposed the Deep Forest (gcForest) model for Android malware detection,

TABLE I
Summary of Deep Learning Papers in Dentistry

DL Method and Ref.	Dataset Used	Pros	Cons	Performance Metrics
Compact Data Learning (CDL) applied to various ML models (RF, KNN, CNN, SVM). [1]	Large datasets with high data complexity-androzoo dataset	Improves efficiency by reducing input features and dataset size while retaining comparable accuracy.	RModels like KNN may suffer from reduced performance or adversarial attacks.	Accuracy rate: Ranges from 95.17% to 99.53%.
Deep Random Forest (DRF) paradigm with malware visualization as grayscale images. Includes ensemble methods, sliding window scanning, and cascade layering. [2]	Malimg	High generalization ability due to ensemble approach.	Computational overhead due to sliding window scanning.	Accuracy rate: 98.65%
MDEB-MVDS-XGB [3]	Behavioral dataset (API calls, network traffic, file access).	Effective against evasive malware	Computationally complex	Accuracy: 99.23%.
Convolutional Neural Network (CNN) for IoT malware detection, converting behavior data into images for training and classification. [4]	Behavior images generated from IoT malware and benign files.	GDetects both known and variant IoT malware.	Requires substantial computational resources for training.	Accuracy rate: 99.87%
Deep Forest (gcForest) combined with CNN and PCA [5]	Android malware dataset	Efficient for small-scale and imbalanced data	Requires significant computational resources	Accuracy rate: Binary Classification: 99.96%.
PlausMal-GAN, GAN framework [6]	Generated and real malware images	Efficient zero-day malware detection	Limited to image-based analysis	Accuracy - 95.56%
MalView, an interactive visual analytics platform [7]	various malware types	Provides detailed visual representation	Limited to visualization	Accuracy: No direct accuracy metric
Hybrid Android malware detection [8]	malware types	Higher detection accuracy	Computationally expensive	Accuracy: Improved
Android Characteristic-Based Method [9]	detecting malware using permissions	Efficient at identifying malware	Limited in handling obfuscated code	Accuracy: 94.5%
CNN-BiLSTM Hybrid Model [10]	Malware samples	Efficient at detecting multivector malware.	High computational complexity.	Accuracy: 96

enhanced with CNN and PCA for feature extraction. This approach demonstrated robust performance, achieving a precision of **99.96%** and a recall of **99.44%**, even for encrypted traffic. Despite its effectiveness, the model demands significant computational resources for training.

Okwon, D., Jang, Y., Lee, S. [6] proposed the PlausMal-GAN framework for zero-day malware detection, leveraging Generative Adversarial Networks (GAN) to generate analogous malware data. The model achieved impressive accuracy, exceeding 99% for zero-day threats. However, it faces limitations due to its computational intensity and reliance on image-based data generation.

Nguyen, H.N., Abri, F., Pham, V., Chatterjee, M., Namin, A.S., Dang, T. [7] introduced MalView, a platform that combines static and dynamic analysis to visualize malware behavior over time. While MalView is not designed as an autonomous detection tool, it effectively aids in identifying suspicious activities through detailed visual analytics. It is particularly valuable for understanding malware actions but does not provide direct detection metrics.

Yunmar, R. A., Kusumawardani, S. S., Widyawan, and Mohsen, F. [8] proposed a Hybrid Android Malware Detection approach that combines static and dynamic analysis to enhance malware detection accuracy. This method leverages the strengths of both techniques, providing better results compared to using either approach individually. While effective in improving detection rates, especially against obfuscated code and hidden behaviors, the hybrid approach is resource-intensive and still faces challenges when dealing with zero-day malware attacks.

Pan, Y., et al. [9] proposed a Hybrid Android Malware Detection approach that combines static and dynamic analysis to enhance the detection of Android malware. This hybrid approach outperforms standalone methods by leveraging the strengths of both techniques, achieving better accuracy but requiring significant computational resources and facing challenges with zero-day attacks. The Android Characteristic-based Method, a static analysis technique, focuses on detecting malware through permissions, API calls, and hardware interactions. It demonstrates high precision (95.8%) and recall (93.4%), making it effective in identifying malware exploiting system vulnerabilities; however, it struggles with obfuscated code and dynamic behaviors. Another method, the Opcode-based Detection, analyzes the opcode sequences of Android applications to identify malware. With an accuracy of 97.58%, this method excels at detecting malware based on execution patterns, though it remains vulnerable to minor code changes commonly used in evasion techniques. Combined, these approaches highlight the potential of static analysis methods while also revealing areas for improvement in handling obfuscation and dynamic threats.

Haq, I. U., Khan, T. A., and Akhunzada, A. [10] proposed the CNN-BiLSTM Hybrid Model for detecting multivector Android malware by combining Convolutional Neural Networks (CNN) and Bidirectional Long Short-Term Memory networks (BiLSTM). This hybrid approach achieves high precision (95%) and recall (94%), outperforming other deep learning models in detecting zero-day malware. The CNN-BiLSTM model leverages the strengths of both architectures, preserving past and future information for improved prediction. While effective, the model is computationally demanding, which can limit its deployment in resource-constrained environments.

Table 1 presents a thorough summary of the latest progress in the field of deep learning for malware detection, which we have identified in our literature review.

VII. CONCLUSIONS

During our analysis,the rapid advancement of malware requires innovative detection methods that go beyond traditional signature-based approaches. Random Forest (RF) has emerged as a powerful machine learning technique for malware detection, offering high accuracy and adaptability. By leveraging multiple decision trees, RF can efficiently classify malicious applications, even in the presence of polymorphic and zero-day threats. It also demonstrates that RF's ensemble learning approach enhances detection rates while reducing false positives, making it a scalable solution for modern cybersecurity challenges. Additionally, RF's ability to process large datasets and perform feature selection helps optimize model performance. When combined with compact data learning (CDL) techniques, RF can maintain high detection accuracy while significantly reducing training time and computational costs.

Despite its effectiveness, RF-based malware detection systems still face challenges, including susceptibility to adversarial attacks and the need for real-time processing efficiency. Future research should focus on integrating RF with explainable AI (XAI) to improve model transparency and developing robust defense mechanisms against evasive malware techniques. Overall, RF presents a promising approach to modern malware detection, offering a balance between accuracy, efficiency, and adaptability in an evolving cybersecurity landscape.

VIII. FUTURE DIRECTIONS

To further improve the effectiveness of Random Forest (RF)-based malware detection, several advancements can be explored. Enhancing adversarial resilience is crucial, as malware creators continuously develop techniques to evade detection. Additionally, integrating Explainable AI (XAI) can improve model transparency by providing insights into feature importance, helping cybersecurity experts better understand and trust detection results. Optimizing RF for real-time malware detection is another key area.Techniques such as feature reduction, parallel processing, and model pruning can help reduce processing time while maintaining accuracy. Furthermore, hybrid approaches that combine RF with deep learning models, such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs), could enhance detection capabilities by leveraging both feature-based and behavioral analysis. Future research should also focus on refining compact data learning (CDL) techniques to further optimize dataset size without compromising performance. By addressing these areas, RF-based malware detection systems can become more effective, scalable, and resilient against sophisticated cyber threats.

REFERENCES

[1] S.-K. Kim, X. Feng, H. A. Hamadi, E. Damiani, C. Y. Yeun, and S. Nandyala, "Advanced machine learning based malware detection systems," IEEE Access, vol. 12, pp. 115296–115305, 2024.

[2] D. A. L. C. Dragos Gavrilut, Mihai Cimpoesu, ""machine learning for reliability engineering and safety ap plications: Review of current status and future opportunities,," vol. 211, 2021.

[3] A. A. Al-Hashmi et al., "Deep-ensemble and multifaceted behavioral malware variant detection model," IEEE Access, 2022.

[4] P.-J. H. Jeon, J. and Y. S. Jeong, "Dynamic analysis for iot malware detection with convolution neural network model," IEEE Access, vol. 8, pp. 96899–96911, 2020.

[5] W.-J. X. J. Zhang, X. and C. Gu, "Detection of android malware based on deep forest and feature enhancement," IEEE Access, vol. 11, pp. 29344–29359, 2023.

[6] D. Okwon, Y. Jang, and S. Lee, "Plausmal-gan: Plausible malware training based on generative adversarial networks for analogous zero-day malware detection," IEEE Transactions on Emerging Topics in Computing, 2023.

[7] H. N. Nguyen, F. Abri, V. Pham, M. Chatterjee, A. S. Namin, and T. Dang, “Malview: Interactive visual analytics for comprehending malware behavior,” IEEE Access, 2022.

[8] R. A. Yunmar, S. S. Kusumawardani, Widyawan, and F. Mohsen, “Hybrid android malware detection: A review of heuristic-based approach,” IEEE Access, 2024.

[9] Y. Pan et al., “A systematic literature review of android malware detection using static analysis,” IEEE Access, 2020.

[10] I. U. Haq, T. A. Khan, and A. Akhunzada, “A dynamic robust dl-based model for android malware detection,” IEEE Access, 2021.