ISSN: 2230-9926

Available online at http://www.journalijdr.com

IJDR

International Journal of Development Research
Vol. 14, Issue, 10, pp. 66863-66868, October, 2024
https://doi.org/10.37118/ijdr.28826.10.2024

RESEARCH ARTICLE OPEN ACCESS

MALWARE ANALYSIS AND DETECTION TECHNIQUES

***[1]Noor Ayesha, [2]Vijayalakshmi, [3]Sherin Nayana B, [3]Shreya, P., [3]Sandhya S and [2]Asha P V**

[1]Assistant Professor, ECE Department, HKBK College of Engineering, Bengaluru, India
[2]Assistant Professor, CSE Department, Atria Institute of Technology, Bengaluru, India
[3]UG Student, CSE Department, Atria Institute of Technology, Bengaluru, India

ARTICLE INFO

ArticleHistory:
Received 19th July, 2024
Received in revised form
14th August, 2024
Accepted 06th September, 2024
Published online 30th October, 2024

Key Words:
Deep learning, Artificial intelligence, Malware, Sandbox evaluation tactics, Engineering techniques.

****Corresponding Author: Noor Ayesha,***

ABSTRACT

As the digital landscape evolves, the sophistication and frequency of malware attacks have escalated, necessitating advanced methodologies for their detection, analysis, and defence. This research delves into state-of-the-art techniques and innovative strategies that are shaping the future of malware combat. We explore the application of deep learning and artificial intelligence in identifying and classifying malware, emphasizing the role of adversarial machine learning in enhancing detection resilience. Behavioural and dynamic analysis methods are scrutinized to uncover malware patterns during execution, alongside the development of countermeasures against sandbox evasion tactics. The study extends to advanced static analysis and automated reverse engineering techniques aimed at expediting the identification of malicious code.

Citation*: *Noor Ayesha, Vijayalakshmi, Sherin Nayana B, Shreya, P., Sandhya S and Asha P V, 2024. *"Malware Analysis and Detection Techniques". International Journal of Development Research*, 14, (10), 66863-66868.

INTRODUCTION

Malware is any type of harmful software that cybercriminals use to harm programmable devices, networks, or services. Malware is often used by cybercriminals to steal data that can be used to gain financial advantage over their victims. According to a recent study malware attacks have been tremendously increasing over the recent years with 6.06 billion attacks worldwide in 2023. This growing rise in Malware attacks has been a cause of concern. Mobile Malware is more commonly found in mobiles that run on Android OS than iOS. Malware in an Android device is generally downloaded through applications. Protection against Malware is one of the most important tasks in Cybersecurity. Different Machine Learning techniques are being developed to detect malware accurately. This has given an opportunity for hackers to come up with more powerful malware. Hence, it is of utmost importance that security researchers be ahead of hackers in order to find a faster and more accurate solution.

History

Year	Mobile evolution	Malware Attack	Detection
1991-2000	1.Consumer Handsets 2.2G Networks 3.GSM standardization 4.Nokia Dominance	1.Timofonica virus (2000) (Not much malware attacks)	1.Rule based systems 2.Statistical Analysis 3.Pattern Recognition
2001-2010	1.Camera and Wap feature 2.Mobile data revolution 3.First touch screen LG Prada	1.Cabir (2004) 2.Commwarrior (2005) 3.FlexiSpy (2005)	1.Decision Trees 2.Random Forests 3.KNN
2011-2020	1.4G connectivity 2.Mobile payments 3.Biometric authentication	1.Phishing 2.Ransomware 3.Spyware and tracking 4.Data breaching	1.Anomly detection 2.Feature extraction 3.Behavioural analysis
2021-2024	1.5G integration 2.Foldable phones 3.AI-ML Learning 4.Security enhancements	1.5G exploitation 2.Privacy breaches 3.Security exploits 4.Supply chain attacks	1.Supervised Learning 2.Deep learning 3.Model Explainability 4.Transfer Learning

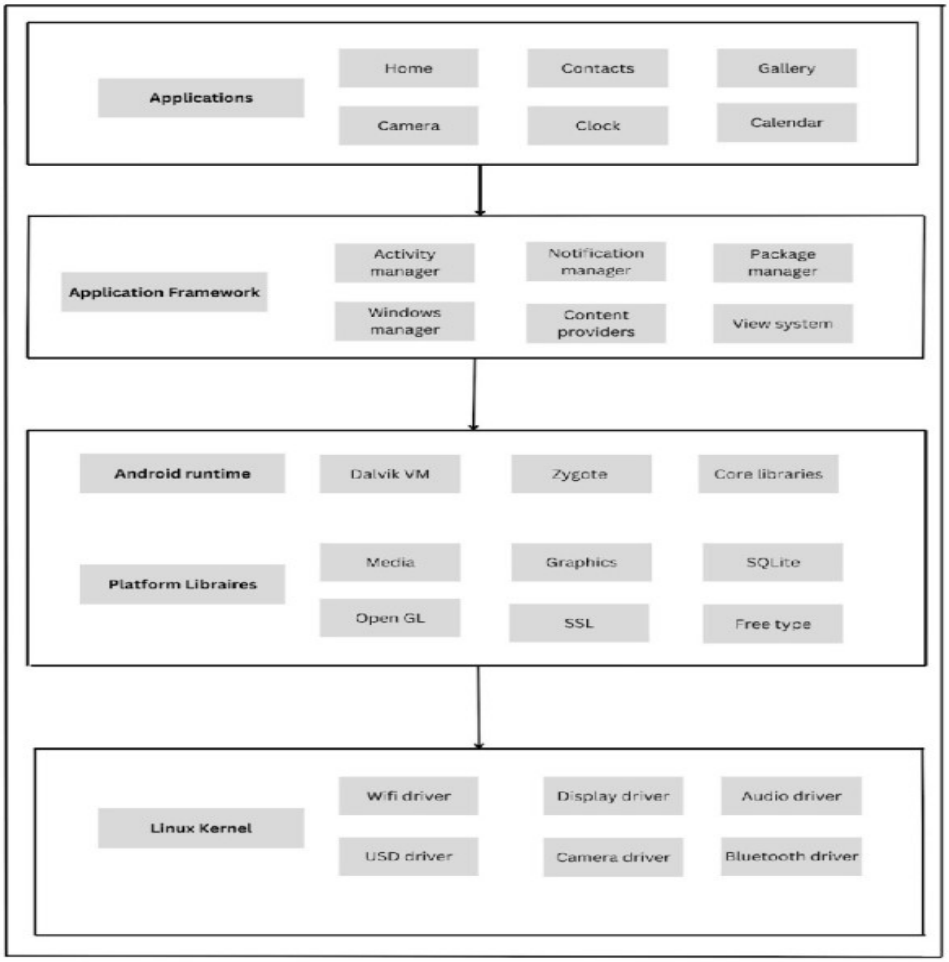
Over the past three decades, mobile technology has advanced from basic handsets to sophisticated smartphones with 5G connectivity. The 1990s marked the start with 2G networks and GSM technology, dominated by Nokia, and malware attacks were rare. Early detection methods relied on rule-based systems and statistical analysis.As technology evolved in the 2000s, with features like cameras and touchscreens becoming standard, mobile malware such as Cabir and FlexiSpy emerged. Detection methods shifted to include decision trees and random forests. The rise of 4G connectivity brought mobile payments and biometric authentication, along with more sophisticated malware attacks like phishing and ransomware. Anomaly detection and behavioural analysis became key in countering these threats. Today, with 5G networks and foldable phones, new security challenges like 5G exploitation and supply chain attacks have emerged. Advanced detection methods now employ deep learning and model explainability, focusing on safeguarding user privacy and data security while supporting technological advancements.

Theoretical Background

Android: Android is a mobile OS based on the modified version of Linux kernel and other open source software.

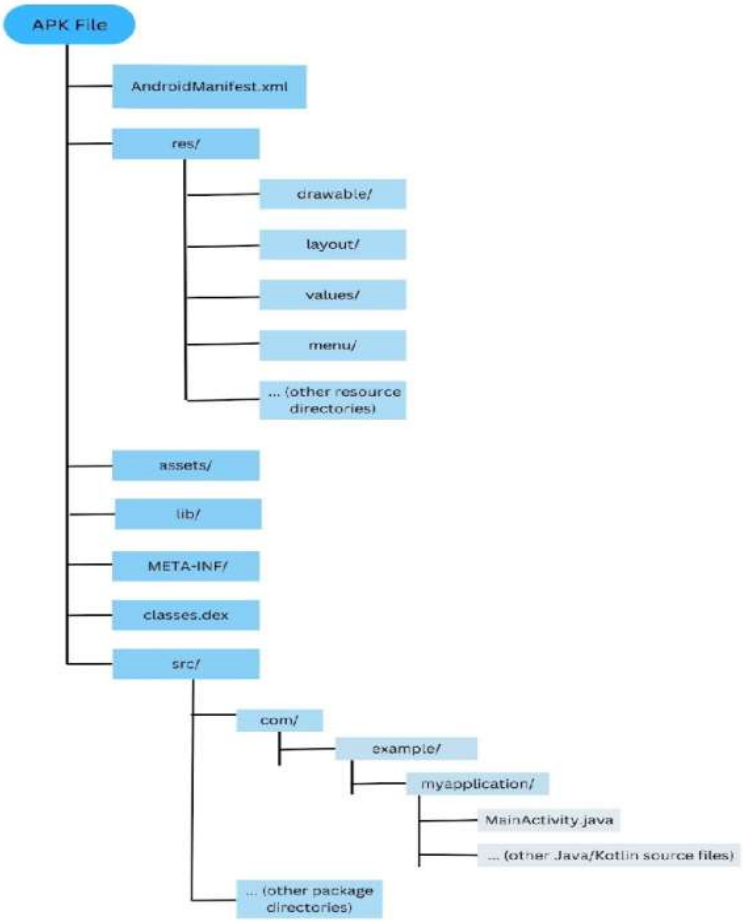
Android Architecture:

1.Applications
2.Application Framework
3.Android runtime and
Platform libraries
4.Linux kernel



A. Linux kernel: Handles security between system and application.
B. Platform Libraries: Includes C/C++ core libraries and java-based libraries such as media, graphic, SOLite, open GL etc to support android development.
C. Android runtime: It's the android runtime environment which is an important part of android. It provides the base for application framework and powers our application with the help of core libraries.
D. Application Framework: It provides generic applications for the hardware access and helps in managing UI with application resources.
E. Application: It's the top layer of android architecture. The pre-installed applications are home, contact, gallery, calendar etc and third party applications, games and all which are installed in this layer only.

Android Application Structure: Understanding the Android package structure is crucial before commencing the reverse engineering of an app. An Android application is encapsulated within an APK file, which comprises various files and directories. Notably, the Android AndroidManifest.xml file contains the app's metadata, including the package name, required permissions, and components such as activities, services, broadcast receivers, and content providers.



Now let us see how android structure helps in malware analysis.The standard layout helps in identifying suspicious apps or applications. The manifest file will contain all the permissions and all, so if any applications requests for unnecessary permission, then it is a sign of malware.The resources directory will contain the resources of apps like image, layouts and all so if any app contains unwanted resources, it can be identified immediately as malware.The assets directory contains raw files so if any unnecessary code or data, it can be detected by examining these files.Library directory contains the necessary libraries and if an application contains unknown libraries, it can be suspicious. Dex files contain compiled code of apps and they will be analysed to check if any harmful code is there or not.With the help of source code directory, experts will be able to analyse the patterns of malware as it contains the apps logic.

Mobile malware analysis: In this section, we provide the summary of the most common types of malwares and how a user can prevent malware attacks on their mobile phones, and tools which are used to detect malware analysis.

Types of mobile malware:

- **Annoying Extras (PUAs):** PUAs often come with free apps and can be tricky to uninstall.
- **Sneaky Stealers (Trojans):** It pretends to be a good app to trick you into installing it, then steals your information or bombards you with ads [8].
- **Hold Your Files Hostage (Ransomware):** Ransomware locks your device or files and won't let you access them until you pay a ransom[8].
- **Secret Spies (Spyware):** Spyware hides in the background, stealing your personal information like passwords, contacts, or what websites you visit[8].
- **Non-Stop Ads (Adware):** Adware throws excessive ads at you, slowing down your device and making it hard to use.
- **Fake Look-Alikes (Fake Apps):** Fake apps pretend to be good apps and after you download, it can be harmful for your device.

How can a user prevent malicious attacks: Regular system updates are crucial as they can prevent around 85% of potential attacks. Setting a strong password and changing it regularly is a good practice. Users should also enable two-factor and multi-factor authentication (2FA and MFA) to enhance security. Avoiding public Wi-Fi and using VPNs adds an extra layer of protection. Device encryption, preferably hardware encryption, is important. It's essential to use only corporate-approved software and create backups regularly. Being cautious is paramount; never open or respond to spam emails and always check the terms and conditions before downloading an application. Using free pop-up blockers and downloading files only from official websites can mitigate risks. Equipping your mobile device with the latest version of reputable software and practicing safe browsing are musts. Installing antivirus software and regularly clearing the browser cache can prevent functional problems and resolve many browser issues.

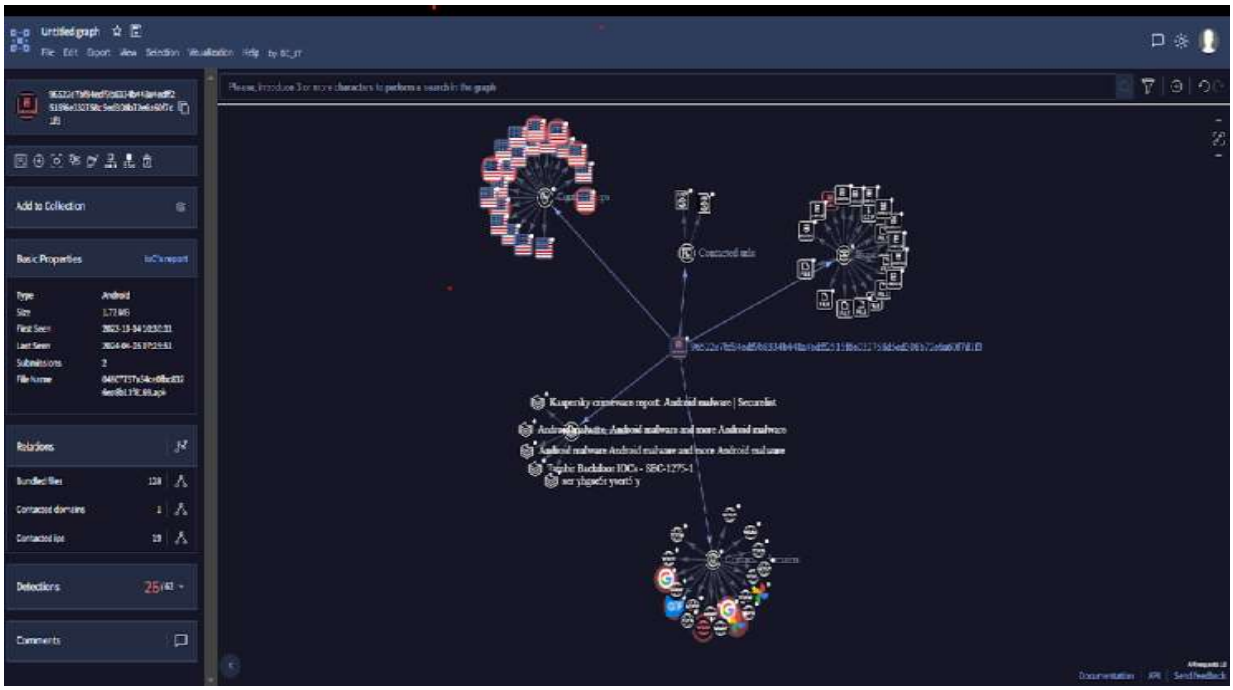
Tools for malware analysis:

1. Virus total
2. Pithus
3. Dropidylsis
4. Medusa

Among these tools, virus total is the one which we are used here to identify malware in an application.

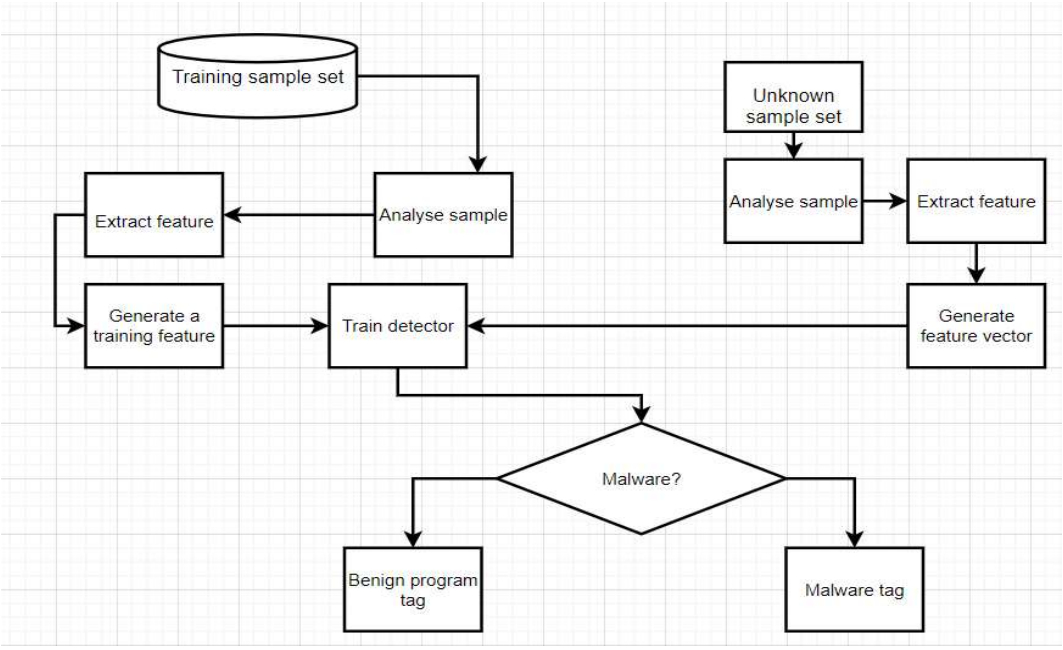
Virus Total: Virus Total is an online platform that scans files and URLs for malicious content using a wide array of antivirus engines. It delivers comprehensive analysis results, leverages community insights, and offers an API for seamless integration with other security tools. The role of these malware analysis tools is to detect, analyse, and understand malicious software, enabling cybersecurity professionals to identify and mitigate threats. The below figure shows us the report of an application or a file which has been scanned by virustotal**.**

These tools provide us with a detailed report which includes relations, description and the IP's connected to them and what kind of malware has been detected.

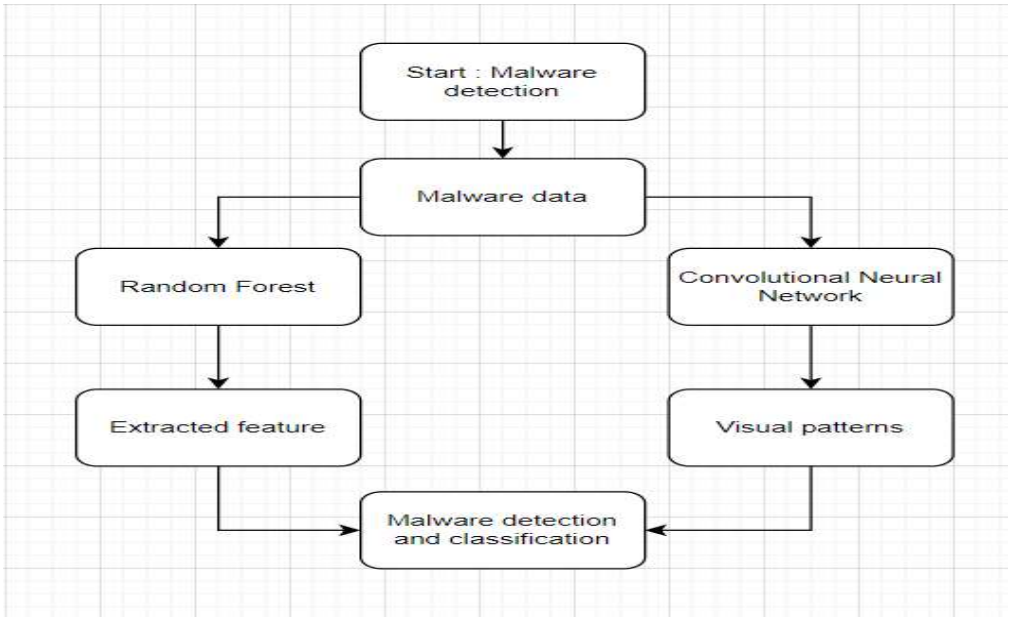


METHODOLOGY

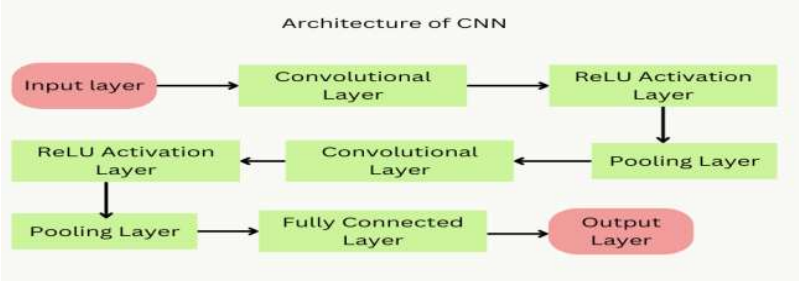
This research paper introduces the various steps and components of a typical machine learning workflow for malware detection and classification, explores the challenges and limitations of such a workflow, and assesses the most recent innovations and trends in the field, with an emphasis on deep learning techniques. To provide a more complete understanding of the proposed machine learning method for malware detection, below is the workflow process from start to finish.



In our research, we are utilizing two machine learning algorithms—Random Forest and Convolutional Neural Network (CNN) to detect and classify malware. Each algorithm brings unique strengths to the table, enhancing the accuracy and robustness of our malware detection system. Random Forest is used for its efficiency in handling large datasets and its ability to rank the importance of features. CNN excels in identifying patterns in visual data, making it ideal for analysing malware represented as images.



***Random Forest*:** Random Forest (RF) is a powerful machine learning algorithm widely used for malware detection, leveraging both static and dynamic data. The process begins by collecting features like byte sequences, PE headers, API calls, and file operations. For static analysis, features such as opcode frequencies and byte histograms are extracted, while dynamic analysis focuses on API call logs. The dataset is preprocessed through cleaning, normalization, and feature selection using techniques like Information Gain. After splitting the data into training and testing sets, typically in an 80-20 ratio, the RF classifier is trained by constructing multiple decision trees. Performance is evaluated using accuracy, precision, recall, F1-score, and k-fold cross-validation. A confusion matrix aids in error analysis. Finally, the trained RF model is deployed in a real-time detection system, with ongoing monitoring and updates to adapt to new threats, ensuring an effective and scalable solution for malware detection



A Convolutional Neural Network (CNN) is a deep learning algorithm well-suited for image recognition tasks, such as malware detection, by learning spatial hierarchies from data. CNNs consist of convolutional layers for feature extraction, pooling layers for dimensionality reduction, and fully connected layers for prediction. In malware detection, pre-processed grayscale images of mobile app usage are analyzed to detect unusual patterns. Combining CNNs with Random Forest (RF) enhances detection accuracy, as RF handles high-dimensional data by constructing multiple decision trees, each making predictions based on unique features. The final result is determined through majority voting, leveraging CNN's feature extraction and RF's classification strengths. This approach provides a robust and scalable solution for malware detection, often outperforming traditional methods with accuracy rates of 90-99%.

RESULT AND PERFORMANCE ANALYSIS

Performance Analysis

Random Forest (RF):

i. **Data Handling:** Capable of handling large sets of data.
ii. **Data Types:** Processes both types of data i.e.,(static and dynamic)
iii. **Metrics:** Uses fine-tuned metrics such as accuracy, precision, F-1
iv. **Accuracy:** Offers a simple yet scalable solution for malware detection.

Convolutional Neural Networks (CNN):

i. **Image Analysis:** It can be used for processing of images and detecting malware.
ii. **Accuracy:** Accuracy of 90-95% can be achieved
iii. **Layers:** Uses Convolutional layers to detect any unusual patterns in visual data.
iv. **Integration:** CNN's feature extraction and RF's classification capabilities.
v. **Performance:** Enhances the overall performance as it uses an integrated system of CNN and RF .
vi. **Scalability:** Provides a scalable solution for malware detection.

Result Summary

i. **Detection Rate:** It can detect malware with a high accuracy of 90-95% in the case of CNN's.
ii. **Feature Extraction:** CNNs excel in extracting features from visual data, while RF ranks feature importance from large datasets.
iii. **Robustness:** The features of CNN as well as RF are unique and robust that have a high chance of providing excellent accuracy.
iv. Combination of RF and CNN gives a superior performance.

CONCLUSION

This research paper presents a comprehensive methodology for malware detection by integrating two distinct machine learning algorithms: Random Forest (RF) and Convolutional Neural Networks

Random Forest excels in handling large datasets and feature importance ranking by constructing multiple decision trees. It processes both static and dynamic data, such as opcode frequencies and API call logs, to provide a scalable and effective malware detection solution. The RF model is fine-tuned using metrics like accuracy, precision, recall, and F1-score, and its performance is validated through k-fold cross-validation and confusion matrix analysis.

Convolutional Neural Networks (CNNs) are highly effective for analysing malware represented as images[7]. CNNs utilize convolutional and pooling layers to learn spatial hierarchies and detect intricate patterns in visual data, achieving high accuracy rates of 90-99%. By integrating CNNs with RF, the system benefits from CNN's powerful feature extraction capabilities and RF's robust classification, creating a scalable and reliable malware detection solution.

PRACTICAL IMPLICATIONS

1. **Improved Mobile Security**: The hybrid detection system offers robust protection against a wide range of malware threats, enhancing overall mobile security for users.
2. **User Trust and Experience**: By minimising false positives and false negatives, the system ensures a better user experience, reducing unnecessary alerts and ensuring that genuine threats are addressed promptly.
3. **Scalability and Adaptability**: The use of machine learning models that continuously learn and adapt to new threats ensure that the detection system remains effective over time, even as malware tactics evolve.

Future Work

1. **Continuous Improvement:** Regular updates to the machine learning models with new data will further improve detection accuracy and adaptability.
2. **Extended Analysis Techniques:** Incorporating additional analysis techniques, such as hybrid cloud-based analysis and advanced heuristic methods, can further enhance the robustness of the detection system.
3. **Real-Time Detection:** Developing real-time detection capabilities to provide immediate protection against malware threats as they arise. In conclusion, the hybrid malware detection methodology for Android mobile devices, combining static, dynamic, behavioural, reputation-based, and machine learning methods, represents a significant advancement in mobile security. Its comprehensive approach, high accuracy, and adaptability to new threats make it a valuable tool in the ongoing effort to protect mobile devices from malware attacks

REFERENCES

1. C. Munoz, "A Brief History of Mobile Malware," Retail Dive, June 2020. A. Kumar, S. Singh, P. Singh, and M. Sharma, "A Comprehensive Survey on Blockchain Technology with Evolution and its Applications," International Journal of Computer Networks and Applications (IJCNA).
2. A. Choudhary, A. Jain, M. Garg, A. N. Singh, and A. Arora, "Detection of Malware Using Deep Learning Techniques," *International Journal of Scientific & Technology Research* (IJSTR).
3. Alzaylaee, M. K., Yerima, S. Y., & Sezer, S. (2017). "DL-Droid: Deep Learning Based Android Malware Detection Using Real Devices."** *Computers & Security*, 70, pp. 72-84.
4. Zhou, Y., & Jiang, X. (2012). "Dissecting Android Malware: Characterization and Evolution."** *IEEE Symposium on Security and Privacy*, pp. 95-109.
5. Sahs, J., & Khan, L. (2012). "A Machine Learning Approach to Android Malware Detection."** *2012- Analyzes the characteristics and evolution of Android malware, providing insights into the challenges of malware detection. European Intelligence and Security Informatics Conference*, pp. 141-147.- Proposes a machine learning approach to detecting Android malware using features extracted from static and dynamic analysis.
6. G. Karabey and A. Aksakalli, "Detection of Android Malware by Using Machine Learning Methods," Communications and Network.
7. A. Choudhary, A. Jain, M. Garg, A. N. Singh, and A. Arora, "Detection of Malware Using Deep Learning Techniques," International Journal of Scientific & Technology Research (IJSTR).
8. M. Cobb, "Mobile Malware," TechTarget, May 2023. - Discusses the implementation of a deep learning-based system for detecting Android malware using real devices.

****