International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 2, Mar-Apr-2026, ISSN (Online): 2395-566X

Hybrid Deep Learning Framework for Android Malware Detection Using Application Permissions and Social Media Threat Intelligence

Mrs.M. Saranya[1], Parimi Sai Neeraja[2], Akumarti Venkat[3], Sistu Sai Purna Sriram[4], Makada Ravikiran[5], Padala Chaitanya[6]

[1]Assistant Professor , [2, 3, 4, 5]B.tech Students Department of CSE, Pragati Engineering College, Surampalem , Andhra Pradesh, India

Abstract- The rapid growth of smartphone usage has made mobile devices an essential part of everyday life, supporting activities such as communication, online banking, education, and social networking. However, the increasing popularity of Android-based devices has also made them a major target for cyber attackers who develop malicious applications to exploit system vulnerabilities and steal sensitive information. To address this challenge, an intelligent malware detection and prevention framework for Android devices is proposed. The proposed system integrates real-time threat intelligence gathered from social media platforms with deep learning-based malware classification techniques. Malware signatures shared through social media sources are periodically collected and stored in a centralized malware hash database to ensure the system remains updated with newly discovered threats. In addition, the system employs a deep learning model based on a Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) architecture to analyze Android application permissions and classify applications as benign or malicious. By combining real-time malware signature updates with deep learning-based behavioural analysis, the proposed framework enhances the accuracy and efficiency of Android malware detection. Experimental evaluation demonstrates that the system achieves high detection accuracy and provides a robust solution for protecting Android devices against emerging malware threats.

Keywords- Android Malware Detection, Deep Learning, Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), Mobile Security, Social Media Threat Intelligence, Tweets Analysis.

I. INTRODUCTION

Smartphones have become an essential part of modern life, supporting a wide range of applications such as communication, online banking, education, social networking, and entertainment. Among the available mobile operating systems, the Android platform holds the largest global market share due to its open-source architecture and flexibility for application development. However, the rapid growth of Android devices has also made them a major target for cybercriminals who develop malicious applications to exploit system vulnerabilities and compromise user data. As a result, ensuring the security and privacy of Android devices has become a significant challenge in mobile computing environments [1].

Android malware poses serious threats to both individual users and organizations. Malicious applications may perform unauthorized activities such as stealing personal information, conducting fraudulent financial transactions, monitoring user behaviour, or remotely controlling infected devices. Many malware applications disguise themselves as legitimate applications and request excessive permissions to gain access to device resources such as contacts, storage, cameras, and network connections. Once installed, these applications can execute harmful operations that compromise the confidentiality, integrity, and availability of user data [2].

Traditional malware detection approaches primarily rely on signature-based detection techniques, where application hashes are compared with known malware signatures stored in security databases. While these methods are effective for identifying previously known malware samples, they often fail to detect new or modified malware variants that evolve continuously to evade detection mechanisms. As cyber threats become more sophisticated, traditional detection systems struggle to identify zero-day malware attacks and advanced persistent threats [6].

To address these challenges, researchers have increasingly adopted machine learning and deep learning techniques for Android malware detection. Machine learning models can analyze large datasets of application features and identify

hidden patterns associated with malicious behaviour. In particular, deep learning architectures such as Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks are capable of learning sequential relationships in application behaviour and detecting suspicious patterns more effectively. These models can analyze application permissions, API calls, and behavioural characteristics to classify applications as benign or malicious with high accuracy [8], [10].

In addition to machine learning techniques, real-time threat intelligence sources such as social media platforms and cybersecurity forums have emerged as valuable resources for identifying newly discovered malware. Security researchers frequently share information about emerging malware families, vulnerabilities, and attack patterns through online platforms. By collecting and analysing such threat intelligence data, security systems can update malware databases more quickly and improve their ability to detect newly emerging threats.

The proposed system introduces a hybrid Android malware detection framework that integrates traditional hash-based detection with deep learning-based classification. The system periodically collects malware signatures from social media sources and updates a centralized malware database to ensure that the latest threats are incorporated into the detection system. Furthermore, a deep learning model based on the RNN-LSTM architecture analyses application permissions and behavioural features to classify Android applications as benign or malicious. By combining real-time threat intelligence with deep learning techniques, the proposed approach enhances malware detection accuracy and provides a more robust defence against evolving Android malware threats.

II. LITERATURE SURVEY

With the rapid growth of Android applications and the increasing number of cyber threats targeting mobile devices, researchers have proposed various techniques for detecting and preventing Android malware. These techniques include machine learning models, deep learning architectures, static analysis, dynamic analysis, and hybrid detection frameworks designed to improve malware detection accuracy and enhance mobile security.

Zhang et al. proposed an Android malware detection approach based on graph attention networks combined with multimodal feature fusion techniques. Their method constructs a specialized call graph structure to analyze both structural and semantic features of Android applications. By integrating permission-based features with code-level information, the proposed system significantly improves malware detection accuracy compared with traditional detection methods [3].

Alamro et al. developed an automated Android malware detection framework using an optimal ensemble learning method. Their model combines multiple machine learning algorithms such as Least Square Support Vector Machines, kernel extreme learning machines, and random vector functional link networks. The study also uses optimization techniques to tune the model parameters, which helps to improve the accuracy and efficiency of malware detection systems [4].

Liu et al. proposed a malware detection method based on a multi-head squeeze-and-excitation residual network architecture. Their approach extracts different static features from Android applications such as permissions, API calls, and hardware components. These features are then processed using a deep neural network that automatically learns important relationships between the features, which helps improve the accuracy of malware classification [5].

Al-Hamiri et al. presented an Android malware detection method based on deep learning using Recurrent Neural Network (RNN) architectures. The proposed model analyzes application behavior and permission features to identify malicious applications. The experimental results showed that deep learning techniques can achieve higher detection accuracy compared to traditional machine learning methods [7].

Jung et al. studied the use of deep neural networks for Android malware detection by analyzing features extracted from Android application packages. Their research showed that deep learning models can learn complex patterns from application data and improve the overall performance of malware detection systems [8].

Sun et al. also examined deep learning-based malware detection using advanced neural network architectures. Their study indicated that deep learning techniques can identify previously unseen malware patterns and improve detection performance when working with large Android malware datasets [10].

From these studies, it can be understood that machine learning and deep learning methods play an important role in improving Android malware detection systems. However, many existing methods mainly depend on static analysis or behavioural analysis techniques, which may face difficulty in detecting newly emerging or rapidly changing malware variants. In addition, most systems do not include mechanisms to integrate real-time threat intelligence information.

Therefore, combining real-time malware signature collection from social media platforms with deep learning-based detection models can improve the capability of malware detection systems. This type of hybrid framework can increase the

adaptability of detection systems and provide better protection against emerging Android malware threats.

III. SYSTEM ANALYSIS

A. Existing System

Traditional Android malware detection systems mainly depend on signature-based detection methods and basic machine learning models. In these systems, Android applications are checked by comparing their cryptographic hash values, such as MD5, SHA-1, or SHA-256, with known malware signatures stored in security databases. If the hash value matches with a known malware signature, the application is identified as malicious. This method works well for detecting already known malware samples, but it cannot effectively detect new or modified malware variants that frequently change to avoid security systems [6].

To solve the limitations of signature-based detection, researchers introduced machine learning-based malware detection systems. These systems study different features taken from Android applications, such as application permissions, API calls, network traffic behavior, and system calls. Various machine learning algorithms like Decision Trees, Support Vector Machines (SVM), Random Forest, Logistic Regression, and Artificial Neural Networks are used to classify applications as either benign or malicious. These models are trained using datasets that contain both malicious and normal applications to improve the detection performance [7], [9].

Recent research has also focused on using deep learning methods for Android malware detection. Deep learning models, especially Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks, can learn complex patterns from application behavior and permission data. These models provide better detection accuracy compared to traditional machine learning methods because they can automatically learn useful features from large datasets [8], [10].

However, many existing malware detection systems still depend on static malware signature databases that need manual updates. Because of this, these systems may not quickly detect newly emerging malware threats. In addition, malware developers often modify malicious applications to escape detection, which makes it difficult for traditional detection systems to identify continuously changing malware variants.

Limitations Of Existing System

- **Limited Detection of New Malware:**

Signature-based detection techniques primarily identify previously known malware and often fail to detect newly emerging or modified malware variants.

- **Static Malware Databases:**

Traditional malware detection systems rely on static signature databases that require manual updates, which limits their ability to respond quickly to emerging threats.

- **High False Positives and False Negatives:**

Conventional machine learning models may incorrectly classify benign applications as malicious or fail to detect sophisticated malware samples.

- **Model Interpretability Issues:**

Advanced machine learning and deep learning models can be difficult to interpret, making it challenging for security analysts to understand detection decisions.

- **High Computational Requirements:**

Deep learning-based detection techniques may require significant computational resources, which can affect performance on mobile devices.

- **Scalability Challenges:**

The rapid growth of Android applications makes it difficult for traditional detection systems to efficiently analyze large volumes of applications.

B. Proposed System

The proposed system presents a hybrid Android malware detection framework that combines real-time threat intelligence with deep learning based classification methods. The main goal of the system is to improve malware detection accuracy and increase the ability to detect both known and unknown malware threats.

In the first stage, the system extracts cryptographic hash values such as MD5, SHA-1, and SHA-256 from Android applications installed on the device. These hash values are compared with malware signatures stored in a centralized malware database. The database is automatically updated every 48 hours by collecting malware hash information shared by cybersecurity researchers on social media platforms using the Twitter API. This automatic update process helps the system receive the latest malware signatures regularly.

In the second stage, if the hash value of an application does not match any known malware signature, the system performs deep learning based classification. The permission information of the application is extracted from the Android manifest file and used as input features for the detection model. A Recurrent Neural Network with Long Short-Term Memory architecture is used to analyze the permission features and classify the application as either benign or malicious.

By combining real-time malware signature updates with deep learning analysis, the proposed system improves the accuracy of Android malware detection and provides stronger protection

against changing malware threats. This hybrid framework increases the adaptability of malware detection systems and helps protect Android devices from both known malware families and newly emerging threats.
.

IV. SYSTEM DESIGN

System Architecture

Below diagram depicts the whole system architecture.

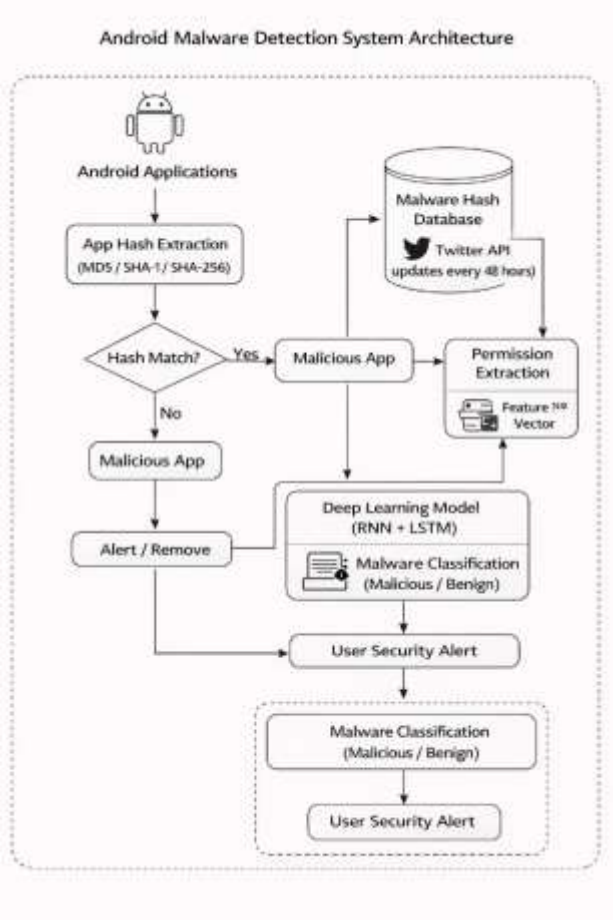

Fig 1. Methodology followed for proposed model

V. SYSTEM IMPLEMENTATION

Modules

This section explains the implementation modules of the proposed Android malware detection framework. The system follows a structured process that includes data collection, feature extraction, malware signature verification, deep learning based classification, and user notification. The modular design helps the system detect malware efficiently and also improves scalability and adaptability to new cybersecurity threats.

A. Data Collection and Preprocessing Module

The data collection module gathers Android applications from different sources to create a dataset that contains both benign and malicious applications. Benign applications are collected from trusted platforms such as official app stores, while malicious applications are obtained from publicly available malware repositories and cybersecurity research datasets.

After collecting the applications, preprocessing is performed to extract useful information needed for malware analysis. The preprocessing process includes removing damaged or corrupted files, validating application packages, and organizing the dataset into a structured format that can be used for feature extraction and machine learning model training.

B. Malware Hash Extraction and Database Update Module

This module extracts cryptographic hash values such as MD5, SHA-1, and SHA-256 from Android application packages. These hash values uniquely identify each application and are compared with malware signatures stored in a centralized malware database.

The malware signature database is automatically updated using the Twitter API. The system collects newly reported malware hash values shared by cybersecurity researchers and security organizations on social media platforms. This automatic update process helps the detection system stay updated with the latest malware signatures and improves its ability to identify newly discovered threats.

C. Permission Feature Extraction Module

In this module, application features are extracted from the AndroidManifest.xml file found inside Android application packages. The manifest file contains detailed information about the permissions requested by an application.

The system analyses permissions such as camera access, contacts access, location access, storage access, SMS services, and network communication. These permission details are converted into a structured dataset that represents the behaviour of each application. The generated feature dataset is used as input for machine learning and deep learning models to perform malware classification.

D. Deep Learning Based Malware Classification Module

The extracted permission features are given as input to a deep learning classification model. The proposed system uses a Recurrent Neural Network with Long Short-Term Memory units to analyze the relationships between application permission features.

The deep learning model learns complex patterns related to malicious applications and classifies each application as either benign or malicious. Compared with traditional machine learning methods, the RNN-LSTM model can better capture hidden relationships between features and detect previously unknown malware patterns.

E. User Notification and Malware Prevention Module

After the classification process, the system generates an alert if a malicious application is detected. The user receives a notification informing them about the possible security risk caused by the detected application.

The system also provides suggestions to the user, such as removing or uninstalling the malicious application. This helps prevent the execution of harmful software on the device and improves the overall security of the system.

VI. RESULTS AND DISCUSSION

This section presents the experimental results and performance evaluation of the proposed hybrid Android malware detection framework. The system combines hash based malware signature detection with a deep learning model that uses a Recurrent Neural Network and Long Short-Term Memory architecture.

Experiments were carried out using datasets that contain both benign and malicious Android applications. Permission features were extracted from the Android applications and used as input for the machine learning and deep learning models. These features help the system analyze the behavior of applications and classify them as benign or malicious.

The evaluation focuses on comparing the performance of different classification algorithms and analyzing the effectiveness of the proposed hybrid malware detection approach.

A. Accuracy Comparison of Machine Learning Models

Several machine learning and deep learning algorithms were evaluated to determine the most effective model for Android malware detection. The evaluated models include Support Vector Machine (SVM), Decision Tree, Random Forest, and the proposed RNN-LSTM model.

Model performance was measured using standard evaluation metrics such as accuracy, precision, recall, and F1-score.

Table 1. Performance Comparison of Malware Detection Models

Model	Accuracy (%)	Precision	Recall	F1-Score
Decision Tree	87.5	0.86	0.85	0.85
Support Vector Machine	89.2	0.88	0.87	0.87
Random Forest	91.4	0.90	0.89	0.89
RNN-LSTM	94.1	0.93	0.92	0.92

From the experimental results, the RNN-LSTM model achieved the highest classification accuracy of 94.1%, outperforming traditional machine learning algorithms. This improved performance is due to the ability of the RNN-LSTM architecture to capture complex relationships between application permissions and malicious behavior patterns.

B. ROC Curve Analysis

The Receiver Operating Characteristic (ROC) curve is used to evaluate the classification performance of the proposed malware detection system by analyzing the trade-off between the True Positive Rate (TPR) and False Positive Rate (FPR).
Fig. 2 illustrates the ROC curve for the proposed Android malware detection model.

.

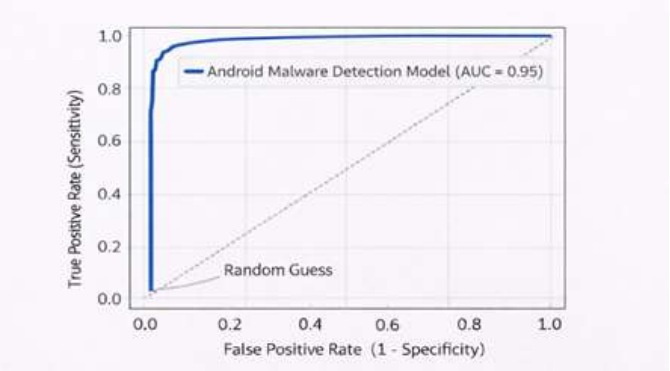

Fig 2. ROC Curve for Android Malware Detection Model

The proposed RNN-LSTM model achieved a ROC–AUC score of 0.95, indicating strong classification capability. A ROC curve positioned closer to the top-left corner of the graph demonstrates that the model can effectively distinguish between malicious and benign applications.

The ROC analysis confirms that the hybrid detection framework provides reliable malware classification performance while maintaining low false positive and false negative rates.

C. Permission Feature Importance Analysis

To understand the contribution of different application permissions to malware detection, a feature importance analysis was conducted on the extracted permission features.

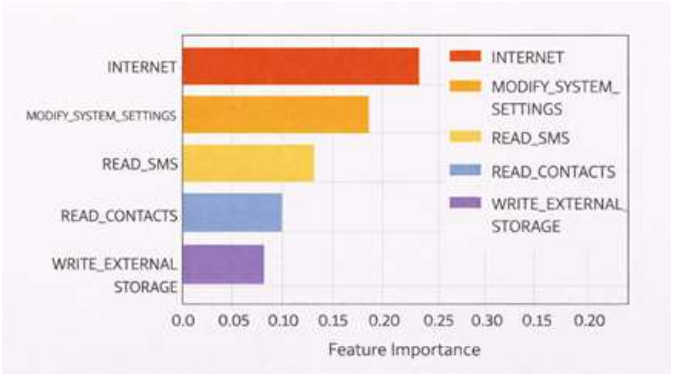

Fig 3. Permission Feature Importance for Android Malware Detection

The analysis revealed that permissions related to internet access, system settings modification, SMS access, contact reading, and external storage access have a significant influence on malware detection. Malicious applications often request excessive or suspicious permissions in order to access sensitive user data or perform unauthorized activities.

The feature importance analysis helps security analysts understand which application permissions are most strongly associated with malicious behavior. This improves the interpretability of the malware detection model and assists in identifying potentially harmful applications.

Overall, the experimental results demonstrate that the proposed hybrid malware detection framework, which integrates real-time malware signature updates with deep learning-based classification, provides improved accuracy and reliability compared with traditional malware detection approaches.

VII. CONCLUSION AND FUTURE WORK

This study presented a hybrid Android malware detection framework that combines traditional hash based detection methods with deep learning based classification. The system uses a Recurrent Neural Network with Long Short-Term Memory architecture to analyze Android application permissions and classify applications as malicious or benign. The system also includes real time threat intelligence by automatically updating the malware hash database using information collected from social media platforms through the Twitter API.

The experimental results show that the proposed framework achieves high detection accuracy and better reliability compared with traditional malware detection methods. The combination of hash based malware signature detection and deep learning analysis allows the system to detect both known malware and newly emerging malware variants effectively. This hybrid method improves the overall security of Android devices by providing a more adaptive and intelligent malware detection system.

In future work, the framework can be improved by adding dynamic behavioural analysis such as monitoring application runtime behavior, system calls, and network traffic patterns. In addition, using advanced deep learning architectures and larger malware datasets may further improve the detection performance. The system can also be extended to support real time malware monitoring in large mobile ecosystems and integrated with cloud based cybersecurity platforms to provide stronger protection for Android devices and mobile networks.

REFERENCES

1. A. McNeil and W. S. Jones, “Malware is surging in Europe: A look at the biggest threats,” Proofpoint Blog, 2022. [Online]. Available: https://www.proofpoint.com/us/blog/email-and-cloud-threats/mobile-malware-surging-europe-look-biggest-threats. Accessed: Jan. 13, 2024.
2. CYBLE, “New SharkBot malware variant discovered (V1.63),” 2022. [Online]. Available: https://blog.cyble.com/2022/02/18/new-sharkbot-variant-discovered/. Accessed: Jan. 12, 2022.
3. X. Zhang, Y. Li, and Z. Wang, “Android malware detection method based on graph attention networks and deep fusion of multimodal features,” Expert Systems with Applications, vol. 235, p. 121617, 2024.
4. H. Alamro, W. Mtouaa, S. Aljameel, A. S. Salama, M. A. Hamza, and A. Y. Othman, “Android malware detection using optimal ensemble learning approach,” Computers & Security, 2023.
5. Y. Liu, H. Zhang, and X. Chen, “Android malware detection based on multi-head squeeze-and-excitation residual networks,” Expert Systems with Applications, vol. 212, p. 118705, 2023.
6. M. A. Al-Hamiri, M. Anbar, and I. H. Hasbullah, “Android malware detection using machine learning techniques,” Procedia Computer Science, vol. 184, pp. 841–846, 2021.
7. M. Al-Hamiri, M. Anbar, and I. H. Hasbullah, “Deep learning-based Android malware detection using static analysis,” Procedia Computer Science, vol. 184, pp. 847–852, 2021.
8. J. Jung, J. Choi, S. Cho, S. Han, and M. Kim, “Android malware detection using deep neural networks,” IEEE Access, vol. 9, pp. 141145–141153, 2021.
9. A. S. Shatnawi, A. Jaradat, T. B. Yaseen, E. Taqieddin, and M. Aland, “Android malware detection based on machine

learning techniques," Wireless Communications and Mobile Computing, Hindawi, 2022.
10. H. Sun, G. Xu, Z. Wu, and R. Quan, "Android malware detection using deep neural networks," Intelligent Automation & Soft Computing, vol. 33, pp. 585–600, 2022.
11. Canadian Institute for Cybersecurity, "Android Malware Dataset (CIC-InvesAndMal2019)." [Online]. Available: https://www.unb.ca/cic/datasets/invesandmal2019.html. Accessed: Jun. 7, 2022.
12. A. B. Singh, "Android malware dataset repository," GitHub, 2022. [Online]. Available: https://github.com/ashishb/android-malware. Accessed: Jun. 7, 2022.
13. Kaggle, "Android malware detection dataset." [Online]. Available: https://www.kaggle.com. Accessed: Jun. 7, 2022.
14. Towards Machine Learning, "Recurrent neural network architecture explained in detail," 2022. [Online]. Available: https://towardsmachinelearning.org/recurrent-neural-network-architecture-explained-in-detail/. Accessed: Sep. 10, 2022.
15. Peltarion, "Binary cross-entropy loss function," 2022. [Online]. Available: https://peltarion.com/knowledge-center/modeling-view/build-an-ai-model/loss-functions/binary-crossentropy/. Accessed: Sep. 10, 2022.