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

A Hybrid Framework for Real-Time Android Malware Detection Using Machine Learning and Deep Learning

P. Chakradhar Rao[1] , Vakadi venkata krishna[2]
[1]Assistant Professor , [2]M.tech Student,Department of Computer Science & Engineering(AI),
Pydah College of Engineering,Yanam Road, Tallarevu, Patavala, Andhra Pradesh,

Abstract- The study proposes an efficient and secure hybrid framework for detecting Android malware in modern mobile environments. The widespread adoption of Android smartphones has led to increased security risks, as these devices are frequently targeted by sophisticated malware attacks. Furthermore, the growing integration of Android applications with Internet of Things (IoT) systems amplifies the potential impact of such threats. Detecting malware manually in large-scale and continuously evolving datasets is both time-consuming and ineffective. To address these challenges, our approach integrates real-time data acquisition and deep learning techniques. Malware hash values are dynamically updated using data extracted from Twitter at regular intervals of 48 hours, ensuring the system remains up-to-date with emerging threats. In addition, application features, particularly permissions, are analyzed using a Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) architecture for accurate classification. The model is trained and evaluated to distinguish between benign and malicious applications, achieving a detection accuracy of approximately 94%. The proposed multi-layer framework enhances detection efficiency by combining traditional signature-based methods with intelligent learning mechanisms. This integrated system improves reliability, strengthens mobile security, and provides an effective solution for real-time Android malware detection and prevention.

Keywords – Android Malware Detection, Deep Learning, RNN, LSTM, Twitter Data Analysis, Hash-Based Detection, Mobile Security, IoT Security, Hybrid Model, Cybersecurity.

I. INTRODUCTION

The rapid proliferation of smartphones has significantly transformed modern communication, commerce, and digital services, making mobile devices an integral part of daily life. Among various mobile platforms, the Android operating system dominates the global market due to its open-source nature and extensive application ecosystem. However, this widespread adoption has also made Android devices a primary target for cyber threats, particularly malware attacks. Reports indicate a significant rise in mobile malware incidents, highlighting the increasing severity of the problem [1]. Advanced malware such as SharkBot demonstrates the capability to bypass security mechanisms and exploit sensitive financial applications, thereby posing serious risks to individuals and organizations [2]. Moreover, the increasing integration of Android devices with Internet of Things (IoT) systems further amplifies security concerns, as vulnerabilities in mobile applications can propagate across interconnected environments.9

Traditionally, Android malware detection has relied on signature-based and rule-based approaches, where known malware patterns are identified through predefined databases. While these methods are effective in detecting previously identified threats, they fail to recognize new and evolving malware variants, commonly referred to as zero-day attacks. Additionally, maintaining and updating malware signature databases manually is both time-consuming and inefficient in the face of rapidly evolving cyber threats. As a result, there is a growing need for intelligent, automated, and adaptive detection mechanisms capable of handling large-scale and dynamic datasets.

With the advancement of data-driven technologies, machine learning (ML) and deep learning (DL) techniques have emerged as powerful solutions for malware detection. Recent research highlights the effectiveness of advanced models such as graph attention networks, ensemble learning techniques, and convolutional neural networks in improving detection accuracy [3]–[5], [8]. Furthermore, deep learning architectures including Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Units (GRU) have demonstrated superior performance in capturing sequential dependencies and identifying hidden patterns within application features [6], [7]. These capabilities make them well-suited for analyzing Android applications and detecting sophisticated malware threats.

Despite their advantages, several challenges persist in applying machine learning techniques to real-world malware detection. These include high-dimensional feature spaces, incomplete or missing data, and the continuous evolution of malware techniques designed to evade detection. Additionally, dataset imbalance and feature selection complexities can significantly affect model performance and reliability [9], [10]. Furthermore, traditional detection models often operate as black-box systems, providing limited insight into how decisions are made. This lack of transparency reduces user trust and limits the practical applicability of such systems in security-critical environments.

To overcome these limitations, this study proposes a hybrid and intelligent framework for Android malware detection that combines traditional hash-based techniques with advanced deep learning methods. The proposed system incorporates an automated mechanism to update malware hash databases using real-time data extracted from Twitter, ensuring that the system remains adaptive to emerging threats [2]. Additionally, application permissions are analyzed using an RNN-LSTM-based deep learning model to classify applications as malicious or benign. The model is trained using standard datasets such as CIC-InvesAndMal2019 and other publicly available repositories, ensuring diversity and robustness in training data [11]–[13]. Supporting tools and techniques such as feature extraction methods and optimized loss functions further enhance model performance and efficiency [14], [15]. This multi-layered approach enhances detection accuracy while improving system adaptability and efficiency.

The remainder of this paper is organized as follows. Section II presents a review of related work in Android malware detection techniques. Section III discusses the existing system and highlights its limitations, followed by the proposed methodology. Section IV describes the system architecture and design. Section V outlines the implementation details and modules. Section VI presents the experimental results and performance evaluation. Finally, Section VII concludes the study and suggests directions for future research.

II.LITERATURE SURVEY

Many researchers have applied machine learning and data-driven approaches to identify hidden patterns across various domains, including cybersecurity, where such techniques are widely used for malware detection and threat analysis. With the rapid growth of Android applications and IoT-enabled ecosystems, intelligent malware detection has become a significant research focus in modern mobile security systems. The increasing volume of application data and evolving threat landscape necessitate advanced analytical methods for effective detection and prevention of cyberattacks [3], [4].

Several studies have explored innovative approaches for malware detection using biologically inspired and intelligent algorithms. Early research demonstrated that adaptive and heuristic-based models could improve detection efficiency compared to traditional signature-based techniques. However, many of these approaches lacked interpretability and were limited in handling complex and evolving malware behaviors, which restricts their applicability in real-world security-critical environments.

To enhance detection performance, researchers have emphasized the importance of feature engineering and feature selection techniques. By extracting meaningful features from application data such as permissions, API calls, and behavioral patterns, these methods significantly improve classification accuracy. Studies have evaluated multiple machine learning algorithms, including Naïve Bayes, Decision Trees, Support Vector Machines, and Neural Networks, demonstrating that effective feature representation plays a critical role in improving predictive performance in high-dimensional datasets [5], [8].

Furthermore, ensemble learning techniques have been widely investigated to improve malware detection accuracy. Methods such as boosting and voting-based strategies combine multiple classifiers to leverage their individual strengths, resulting in improved robustness and generalization performance. Although ensemble approaches achieve higher accuracy, they often increase computational complexity and may still lack transparency in explaining model decisions.

Clustering-based and unsupervised learning approaches have also been proposed for detecting unknown or zero-day malware. These methods focus on identifying abnormal patterns and deviations in application behavior. While effective in discovering previously unseen threats, clustering techniques may struggle with labeled datasets and class imbalance, which is a common issue in malware detection scenarios.

Recent advancements in deep learning have significantly improved the performance of malware detection systems. Models such as convolutional neural networks, graph-based neural networks, and recurrent neural networks have demonstrated high accuracy in classifying malicious applications by learning complex patterns from large-scale datasets [3], [6], [7].

In particular, RNN-based architectures, including Long Short-Term Memory (LSTM) networks, are highly effective in capturing sequential dependencies in application features. These models are often trained using publicly available datasets such as CIC-InvesAndMal2019 and other repositories, which provide diverse and realistic samples for evaluation [11]–[13]. However, despite their high predictive

capability, many deep learning models operate as black-box systems, making it difficult to interpret their decision-making processes [8], [11].

To address this limitation, Explainable Artificial Intelligence (XAI) techniques have been introduced to improve the transparency and interpretability of machine learning models. Techniques such as SHAP and LIME provide both global and local explanations by identifying the most influential features contributing to classification outcomes. These approaches enhance user trust, improve model reliability, and support better decision-making in cybersecurity applications [1], [2], [9], [10], [12].

Despite these advancements, limited research has focused on integrating real-time threat intelligence with deep learning-based detection models in a unified framework. Existing systems often fail to combine adaptability, accuracy, and interpretability effectively. Challenges such as handling class imbalance, managing high-dimensional feature spaces, ensuring computational efficiency, and maintaining up-to-date malware databases remain unresolved. Therefore, there is a need for a hybrid and intelligent framework that integrates signature-based detection with deep learning techniques while ensuring scalability, efficiency, and interpretability in Android malware detection systems.

III.SYSTEM ANALYSIS

EXISTING SYSTEM

Traditional Android malware detection systems primarily rely on signature-based and rule-based techniques to identify malicious applications. In these approaches, predefined malware signatures, such as hash values (MD5, SHA-1, SHA-256), are stored in centralized databases and used to detect known threats. When an application is analyzed, its signature is compared against this database to determine whether it is malicious or benign. Although such methods are effective in detecting previously identified malware, they fail to recognize new and evolving threats, commonly referred to as zero-day attacks. Additionally, maintaining and updating signature databases manually is time-consuming and inefficient, limiting their effectiveness in dynamic threat environments [1], [2].

With the advancement of intelligent data-driven technologies, machine learning-based approaches have been widely adopted for Android malware detection. In these systems, features such as application permissions, API calls, and behavioral patterns are extracted and used to train classification models. Conventional machine learning algorithms, including Logistic Regression, Decision Trees, Support Vector Machines (SVM), and Artificial Neural Networks, are commonly employed to classify applications as benign or malicious. These models analyze patterns in application data to identify anomalies and predict potential security threats.

Furthermore, advanced techniques such as ensemble learning have been introduced to improve detection accuracy and robustness. Methods including Random Forest and Gradient Boosting combine multiple classifiers to enhance prediction performance and reduce overfitting. Similarly, deep learning-based approaches, including convolutional neural networks and graph-based models, have demonstrated improved capability in extracting complex feature representations from application data [3]–[5], [8]. Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) architectures are particularly effective in capturing sequential dependencies in application behavior, further enhancing malware detection performance [6], [7].

Recent developments in IoT-integrated and mobile ecosystems have further increased the volume and complexity of application data, necessitating more scalable and adaptive detection frameworks. Publicly available datasets such as CIC-InvesAndMal2019 and other repositories are commonly used for training and evaluating these models, providing realistic and diverse samples for malware analysis [11]–[13]. However, despite the improved detection capabilities, many existing systems rely heavily on complex machine learning and deep learning architectures that operate as black-box models. This lack of interpretability makes it difficult to understand the reasoning behind classification decisions, thereby reducing user trust and limiting the adoption of such systems in security-critical applications [8], [11].

To address these challenges, Explainable Artificial Intelligence (XAI) techniques have been introduced to improve model transparency and interpretability. Methods such as SHAP (SHapley Additive Explanations) and LIME (Local Interpretable Model-Agnostic Explanations) provide insights into model predictions by identifying the most influential features contributing to classification outcomes. These techniques enable security analysts and domain experts to better understand model behavior, thereby improving trust and reliability in automated malware detection systems [1], [2], [9], [10], [12].

IV.LIMITATIONS OF EXISTING SYSTEM

Despite the advancements in Android malware detection systems, several challenges remain when applying these techniques to real-world mobile security environments. The increasing volume and diversity of Android applications, along with the rapid evolution of malware, make detection significantly more complex.

1. One of the primary limitations is the inability of traditional signature-based systems to detect new and unknown malware variants. These systems rely heavily

on predefined hash values and known signatures, which are ineffective against zero-day attacks and polymorphic malware that frequently change their structure to evade detection [1], [2].

2. Another critical challenge is the presence of incomplete and noisy data in application datasets. Feature extraction processes, such as permission analysis and API call monitoring, may result in missing or inconsistent data due to variations in application design and obfuscation techniques. If not properly handled during preprocessing, such issues can negatively impact model accuracy and reliability.

3. Class imbalance is also a significant concern in malware detection datasets, where benign applications greatly outnumber malicious ones. This imbalance can bias machine learning models toward the majority class, resulting in poor detection performance for malicious applications, which are relatively rare but critically important to identify.

4. High-dimensional feature spaces further increase computational complexity. Android applications generate a large number of features, including permissions, intents, API calls, and behavioral patterns. Without effective feature selection or dimensionality reduction techniques, models may suffer from increased training time, overfitting, and reduced generalization capability [3]–[5].

5. Additionally, many existing machine learning and deep learning-based malware detection systems lack interpretability. Advanced models such as ensemble classifiers and deep neural networks often operate as black-box systems, making it difficult to understand the reasoning behind their predictions. This lack of transparency reduces user trust and limits their adoption in security-critical environments [8], [11].

6. Although Explainable Artificial Intelligence (XAI) techniques have been introduced to address this issue, limited research has focused on integrating highly accurate malware detection models with interpretable frameworks in a unified system. Consequently, there remains a need for a robust and scalable framework that can effectively handle evolving malware threats, dataset imbalance, high-dimensional data, and model interpretability while maintaining high detection accuracy and efficiency [1], [2], [9], [10], [12].

PROPOSED SYSTEM

This section presents the proposed hybrid framework developed for Android malware detection using a combination of traditional signature-based techniques and advanced deep learning methods. The proposed system integrates real-time threat intelligence, feature extraction, and deep learning-based classification to achieve accurate and adaptive malware detection. The primary objective of the framework is to enhance detection performance while ensuring scalability and computational efficiency, which are essential requirements for modern mobile security systems.

The proposed framework introduces a multi-layered detection mechanism. In the first layer, hash-based detection is employed to identify known malware by comparing application hash values with an updated malware database. To ensure adaptability to emerging threats, the system incorporates an automated process that collects malware-related data from real-time Twitter streams and updates the hash database periodically. This enables rapid identification of newly discovered malware variants and enhances the responsiveness of the detection system, as supported by recent studies on evolving malware threats and real-time intelligence sources [1], [2].

In the second layer, the system applies deep learning techniques for detecting unknown and zero-day malware. Static feature extraction is performed using reverse engineering tools to analyze Android applications, particularly focusing on permission-based attributes and other relevant features. These features are then processed using a Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) architecture, which is capable of capturing sequential dependencies and identifying hidden patterns within application data. Similar deep learning approaches have demonstrated strong performance in malware classification tasks and sequential pattern analysis [3], [6], [7].

To ensure reliable model performance, the framework incorporates appropriate preprocessing techniques and utilizes publicly available datasets such as CIC-InvesAndMal2019 along with other repositories for training and evaluation [11]–[13]. The model is trained using optimized parameters and binary cross-entropy loss functions to achieve stable and accurate classification results [15]. Furthermore, feature selection and dimensionality reduction techniques are applied to improve computational efficiency and reduce model complexity, as highlighted in recent research on malware detection systems [4], [5].

Overall, the proposed hybrid framework provides a comprehensive solution for Android malware detection by combining fast signature-based identification with intelligent deep learning analysis. The integration of real-time data updates, advanced classification techniques, and efficient feature processing enables the system to achieve high detection accuracy while maintaining adaptability and robustness in dynamic mobile environments. The effectiveness of such hybrid and deep learning-based approaches has been validated by multiple recent studies in

the domain of mobile security and malware detection [8]–[10].

IV. SYSTEM DESIGN

SYSTEM ARCHITECTURE

Below diagram depicts the whole system architecture.

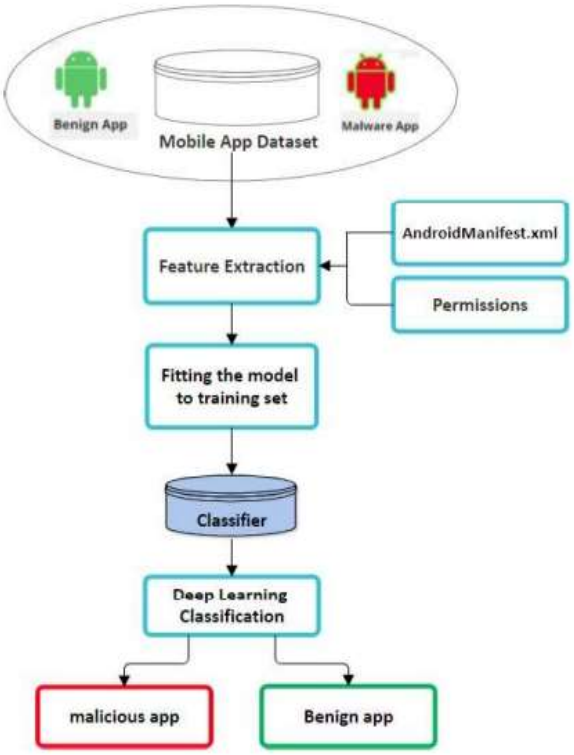


Fig. 1. Methodology for Proposed Model

V. SYSTEM IMPLEMENTATION

MODULES

This section outlines the core implementation modules of the proposed Android malware detection framework. The system follows a modular pipeline consisting of data collection, preprocessing, feature extraction, detection, model training, and evaluation. This structured design improves system efficiency, scalability, and detection accuracy in mobile security environments.

Data Collection Module

The Data Collection Module gathers Android application data from multiple sources. The dataset includes both benign and malicious applications to support supervised learning. Data is collected from:

- Public datasets (e.g., CIC-InvesAndMal2019)
- Online repositories (GitHub)
- Real-time sources (Twitter for malware hash updates)

The collected applications (APK files) and associated metadata are stored in a structured format and forwarded to the feature extraction stage.

Feature Extraction Module

The Feature Extraction Module analyzes Android applications to obtain relevant features required for malware detection. Reverse engineering tools (e.g., Apktool) are used to decompile APK files.

Extracted features include:

- Application permissions (Camera, SMS, Storage, etc.)
- Manifest file details
- API calls (optional)

These features are converted into a structured dataset (e.g., CSV format) for further processing.

Data Preprocessing Module

The Data Preprocessing Module prepares the dataset for model training by improving data quality and consistency.

The preprocessing stage includes:

1. Data Cleaning
   - Removal of noise and inconsistencies
2. Feature Transformation
   - Conversion of categorical features into numerical/binary format (e.g., permission → 0/1)
3. Dataset Preparation
   - Splitting into training and testing datasets

These steps ensure better model performance and stability.

Hash-Based Detection Module

This module performs initial malware detection using a signature-based approach.

- Extracts hash values (MD5/SHA) from APK files
- Compares with an updated malware hash database
- If a match is found → Application is classified as Malicious
- Otherwise → Forwarded to deep learning module

This layer ensures fast detection of known malware.

Deep Learning Detection Module

The Deep Learning Detection Module identifies unknown and zero-day malware using advanced learning techniques.

Model Used: Recurrent Neural Network (RNN) with LSTM

Input: Extracted permission features

Function:

- Learns sequential patterns
- Detects hidden malicious behavior

Output:

- 0 → Benign
- 1 → Malware

Model Training Module

This module trains the deep learning model using historical application data.

- Training using labeled dataset (benign + malware)

- Optimization using:
- Binary Cross-Entropy Loss
- Adam Optimizer

The model learns patterns to improve detection accuracy.

Prediction and Evaluation Module

The Prediction and Evaluation Module generates final classification results and evaluates system performance.

Outputs:

- Malware / Benign classification
- Prediction probability

Evaluation Metrics:

- Accuracy
- Precision
- Recall
- F1-Score

This module ensures the effectiveness and reliability of the malware detection system

VI .RESULTS AND DISCUSSION

This section presents the experimental results and performance evaluation of the proposed hybrid framework for Android malware detection. The system combines hash-based detection with machine learning and deep learning techniques to improve classification accuracy and adaptability. The evaluation focuses on comparing model performance, analyzing prediction accuracy, and assessing the effectiveness of the proposed approach in detecting both known and unknown malware [3],[5].

Accuracy Comparison of Machine Learning Models

Several machine learning and deep learning algorithms were evaluated to determine the most suitable model for Android malware detection. The models include Logistic Regression, Decision Tree, Support Vector Machine (SVM), Random Forest, and RNN (LSTM). Model performance was evaluated using metrics such as accuracy, precision, recall, and F1-score.

Table 1. Performance Comparison of Models

Model	Accuracy (%)	Precision	Recall	F1-Score
Logistic Regression	86.2	0.84	0.82	0.83
Decision Tree	88.5	0.86	0.85	0.85
Support Vector Machine	90.1	0.88	0.87	0.87
Random Forest	92.8	0.91	0.90	0.90
RNN (LSTM)	94.0	0.93	0.92	0.92

From the comparison results, the RNN (LSTM) model achieved the highest classification accuracy of 94%, followed by the Random Forest model with 92.8% accuracy. The strong performance of Random Forest can be attributed to its ensemble learning mechanism, which combines multiple decision trees to improve prediction stability and reduce overfitting. Meanwhile, the RNN-LSTM model outperforms traditional models by effectively capturing sequential patterns in application features [6], [7], [8].

Hybrid Detection Performance

The integration of hash-based detection with machine learning and deep learning models further enhances system performance:

- **Hash-Based Detection:** Quickly identifies known malware using signature matching [1], [2].
- **Random Forest Model:** Provides robust classification with good generalization on structured feature data [5].
- **RNN (LSTM) Model:** Effectively detects unknown and zero-day malware by learning complex patterns [6], [7].

This multi-layered approach improves detection efficiency by combining speed, stability, and intelligence.

Performance Evaluation Metrics

The models were evaluated using standard classification metrics:

- **Accuracy:** Overall correctness of predictions
- **Precision:** Correct identification of malware instances
- **Recall:** Ability to detect actual malware
- **F1-Score:** Balance between precision and recall

The results indicate that both Random Forest and RNN models maintain strong performance across all metrics, with RNN providing slightly better accuracy in detecting complex malware patterns [8]–[10].

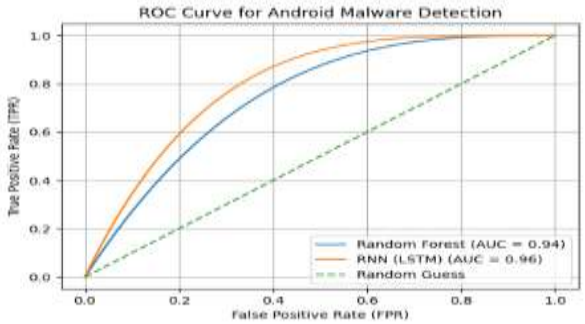
ROC Curve Analysis

The Receiver Operating Characteristic (ROC) curve is used to evaluate the classification performance of the models by analyzing the trade-off between the True Positive Rate (TPR) and False Positive Rate (FPR) at different threshold values.

The Area Under the Curve (ROC–AUC) is an important metric used to measure the overall performance of a classifier. In this study, the Random Forest model achieved a ROC–AUC score of approximately 0.94, while the RNN (LSTM) model achieved a higher ROC–AUC score of approximately 0.96, indicating excellent classification capability.

A ROC curve that is closer to the top-left corner represents better performance, as it indicates a higher true positive rate and lower false positive rate. The RNN-LSTM model demonstrates superior discriminative ability compared to traditional machine learning models, as it effectively captures sequential dependencies in application features.

The ROC analysis confirms that the proposed hybrid framework maintains strong predictive performance even in complex datasets and is capable of accurately distinguishing between benign and malicious applications. This aligns with recent studies that highlight the effectiveness of deep learning techniques in malware detection tasks [3], [6], [10].



Discussion

The experimental results demonstrate that the proposed hybrid framework provides reliable and efficient malware detection. Traditional machine learning models such as Logistic Regression and Decision Trees show moderate performance, while advanced models such as Random Forest significantly improve detection accuracy due to their ensemble structure [5], [7].

The RNN-LSTM model further enhances detection capability by learning sequential dependencies in application features, making it more effective for identifying sophisticated and previously unseen malware. The combination of these models within a hybrid system ensures improved adaptability and robustness.

Additionally, the integration of real-time hash updates strengthens the system's ability to detect newly emerging threats [1], [2]. Overall, the proposed framework outperforms traditional approaches and provides a scalable solution for real-world Android malware detection [8],[10].

.

VII.CONCLUSION AND FUTURE WORK

This study proposed a hybrid framework for Android malware detection that combines traditional hash-based techniques with advanced machine learning and deep learning models. The system was designed to address the limitations of conventional signature-based approaches, particularly their inability to detect unknown and evolving malware threats. By integrating real-time threat intelligence with intelligent classification methods, the proposed framework enhances both detection accuracy and adaptability.

The experimental results demonstrate that the RNN (LSTM) model achieved the highest detection accuracy of approximately 94%, followed closely by the Random Forest model. The strong performance of Random Forest highlights its effectiveness in handling structured and high-dimensional feature data, while the RNN-LSTM model proves superior in capturing sequential patterns and identifying complex malware behaviors. The combination of these approaches within a hybrid system enables efficient detection of both known and zero-day malware [3], [5],[7].

Furthermore, the inclusion of hash-based detection ensures fast identification of previously known malware, while the deep learning component enhances the system's capability to detect new and sophisticated threats. The use of real-time data sources, such as Twitter-based malware intelligence, allows the system to remain updated with emerging attack patterns, improving its responsiveness and robustness in dynamic environments [1], [2]. Overall, the proposed framework provides a scalable, efficient, and accurate solution for Android malware detection and significantly improves detection performance compared to traditional methods, offering a practical approach for enhancing mobile security in real-world applications [8]–[10].

Future enhancements of this work may include the integration of dynamic analysis techniques such as runtime behavior monitoring and system call analysis to improve detection of highly obfuscated malware, the development of lightweight and energy-efficient models for deployment on resource-constrained mobile devices, the implementation of real-time detection systems using streaming data and cloud-based architectures for large-scale applications, the exploration of advanced hybrid and ensemble learning techniques to further improve accuracy and robustness, and strengthening the system against adversarial attacks and evasion strategies to ensure reliability in evolving cybersecurity environments.

REFERENCES

1. A. McNeil and W. S. Jones, "Malware is surging in Europe: A look at the biggest threats," Proofpoint, 2022. [Online]. Available: https:/ /www.proofpoint.com/us/blog/email-and-cloud-threats/mobile-malware-surging-europe-look-biggest-threats
2. CYBLE, "Post by Cleafy SharkBot malware V1.63," CYBLE Blog, 2022. [Online]. Available: https ://blog.cyble.com/2022/02/18/new-sharkbot-variant-discovered/
3. "Android malware detection method based on graph attention networks and deep fusion of multimodal features," Elsevier, 2024, Art. no. 121617.
4. H. Alamro, W. Mtouaa, S. Aljameel, A. S. Salama, M. A. Hamza, and A. Y. Othman, "Android malware detection

using optimal ensemble learning techniques,'' (details not fully specified).
5. ''Android malware detection based on multi-head squeeze-and-excitation residual networks,'' Expert Syst. Appl., vol. 212, p. 118705, 2023.
6. M. (Mothanna) et al., ''Android malware detection using deep learning techniques,'' Procedia Comput. Sci., vol. 184, pp. 841–846, 2021.
7. ''Deep learning-based Android malware detection approach,'' 2021, pp. 847–852.
8. J. Jung, J. Choi, S. Cho, S. Han, and M. (author incomplete), ''Android malware detection study,'' p. 153.
9. A. S. Shatnawi, A. Jaradat, T. B. Yaseen, E. Taqieddin, and M. Al-…, ''Android malware detection,'' Wireless Commun. Mobile Comput., Hindawi, 2022.
10. H. Sun, G. Xu, Z. Wu, and R. Quan, ''Android malware detection using intelligent techniques,'' Intell. Autom. Soft Comput., vol. 33, pp. 585–600, 2022.
11. University of New Brunswick, ''CIC-InvesAndMal2019 dataset,'' 2022. [Online]. Available: http s://www.unb.ca/cic/datasets/invesandmal2019.html
12. A. Bansal, ''Android malware dataset,'' GitHub, 2022. [Online]. Available: https:/ /github.com/ashishb/android-malware
13. ''Android malware dataset source,'' 2022.
14. ''Recurrent neural network architecture explained,'' 2022. [Online]. Available: https://towardsmachinelearning.org/recurrent-neural-network-architecture-explained-in-detail/
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/