IJIRAE:: International Journal of Innovative Research in Advanced Engineering ISSN: 2349-2163
Volume 12, Issue 12, December 2025 https://www.ijirae.com/archives
https://doi.org/10.26562/ijirae.2025.v1212.02

An Effective Machine and Deep learning Algorithms for Android Malware Classification and Detection

Dr.Moses Praveen Mukherjee
Professor, Department of Computer Applications,
East Point College of Higher Education, Bengaluru, India
dr.moses@eastpoint.ac.in

Publication History
Manuscript Reference No: IJIRAE/RS/Vol.12/Issue12/DCAE10081
Research Article | Open Access | Double-Blind Peer-Reviewed | Article ID: IJIRAE/RS/Vol.12/Issue12/DCAE10081
Received:20, November 2025,Revised:29,November 2025,Accepted: 01,December 2025,Published Online:12,December 2025. https://www.ijirae.com/volumes/Vol12/iss-12/02.DCAE10081.pdf
Article Citation:Dr.Moses(2025),An Effective Machine and Deep learning Algorithms for Android Malware Classification and Detection, IJIRAE: International Journal of Innovative Research in Advanced Engineering, Volume 12, Issue 12 of 2025 pages 706-709 **Doi:>** https://doi.org/10.26562/ijirae.2025.v1212.02
BibTeX Key: Dr.Moses@2025Effective
IJIRAE papers should be cited as IJIRAE (International Journal of Innovative Research in Advanced Engineering, AM Publications, India 2025, ISSN 2349-2163, https://doi.org/10.26562/ijirae.2025.v1212.02 The journal's official abbreviation is IJIRAE. Orcid: https://orcid.org/0009-0004-9398-7488

Abstract: The proliferation of Android malware has become a significant concern in the cyber security landscape. Traditional signature-based detection methods are no longer effective against the rapidly evolving malware threats. Machine Learning (ML) and Deep Learning (DL) algorithms have emerged as a promising solution for Android malware classification and detection. This study aims to investigate the effectiveness of various ML and DL algorithms for Android malware detection. This work analyses the performance of several algorithms, including Support Vector Machines (SVM), Random Forest (RF), and Recurrent Neural Network (RNN). From this analysis, reveals the DL algorithms, particularly RNN, outperform traditional ML algorithms in terms of accuracy, precision, and recall. This finding suggests that the DL algorithm and selects the features can provide an effective solution for Android malware classification and detection. The results of this study can be used to develop a robust and efficient Android malware detection system, which can help protect against the increasing threats of mobile malware. Overall, this study demonstrates the potential of ML and DL algorithms in detecting Android malware and provides insights into the development of effective detection systems.
Keywords: Android malware, Machine Learning, Deep Learning, Support Vector Machine (SVM), Random Forest (RF), Recurrent Neural Network (RNN), Android classification and detection.

1. INTRODUCTION

Google Android dominates the smartphone market with a 75% market share and 2.8 billion active users [1]. The open-source nature and customizability of Android have made it a popular choice among users and manufacturers, leading to the production of low-cost smart devices. Additionally, Android's Software Development Kit (SDK) has made it easy for developers to create Android applications, making it a large target market. However, this popularity has also led to an increase in malicious activity targeting Android devices through malware apps. According to AV-Test, the number of total malware has increased from 182 million to 1342 million in the last ten years [2]. The primary goal of malware apps is to compromise user privacy and sensitive data by performing unwanted actions. To address this issue, Android malware detection using machine learning has emerged as a promising approach [3]. One effective way to identify suspicious applications is by monitoring the network to which the Android device is connected. Machine learning, a subset of artificial intelligence, enables the development of computer programs that can learn from data and improve their performance over time. This technology can be used to build models that evaluate incoming data to make predictions and detect anomalies, which can be applied to detect malicious Android applications [4]. Researchers have explored various features, such as permissions requested, API calls made, and network activity, to build models that can classify applications as malicious or non-malicious. Studies have shown that Random Forest and Ensemble techniques achieve high accuracy in malware detection [5]. Performance evaluation is typically done using 2-fold cross-validation, F1-score, Precision, and Recall. The increasing threat of Android malware has highlighted the need for analyzing prominent malware samples [6] [7]. Researchers have made significant efforts in static and dynamic malware analysis using static features and API calls, respectively [8]. A study demonstrated that detecting harmful traffic on computer systems and improving network security is possible by employing machine learning algorithms, such as Naive Bayes, SVM, J48, RF, and a proposed approach [9]. Malware detection modules are responsible for analyzing collected data and determining whether a specific piece of software or network connection poses a security concern [10] [11].

Machine learning systems can explicitly express the principles underlying the patterns they have observed, enabling algorithms to improve their predictive ability using feedback from previous tasks [12]. By leveraging machine learning, it is possible to develop effective malware detection systems that can protect Android devices from malicious threats [14].

2. METHODOLOGY

Figure 1 the malware detection system's process begins with dataset preparation, which involves collecting and organizing a malware dataset for analysis. During the preprocessing step, raw data is converted into a format suitable for machine learning models using techniques such as label encoding, which turns categorical labels in to numerical values. After the data is preprocessed, the model construction entails training various ML algorithms such as SVM, RF, and RNN to learn patterns that distinguish malicious files from begin ones. After the models have trained, the next step is to evaluate their performance using measures such as accuracy, recall, F1-score and precision. These assessment criteria assist establish how preciously and effectively the models identify malware.

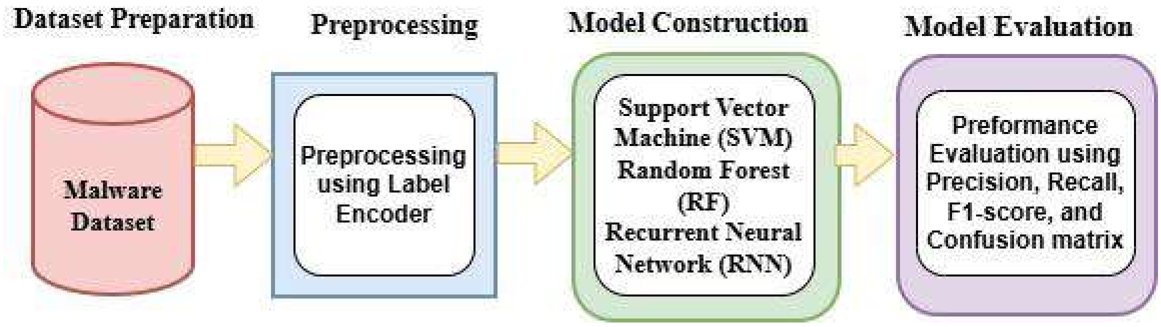

Fig 1 Overall Proposed Methodology

Dataset Preparation

The malware dataset used in this work is sourced from Kaggle, a widely used platform for publicly available machine learning datasets. The Kaggle malware dataset contains a large collection of labeled samples representing both malicious and benign files. Each sample includes extracted features such as opcode frequencies, byte-level patterns, API call sequences, and other static or dynamic attributes essential for malware classification. The dataset provides clear labels that categorize each instance as malware or non-malware, and in some cases further classifies malware into multiple families. Because the dataset is already structured and well-organized, it allows for efficient preprocessing, including label encoding and feature normalization. This Kaggle dataset serves as a reliable benchmark for training and evaluating machine learning models aimed at malware detection.

3. PREPROCESSING USING LABEL ENCODER

In the preprocessing stage, a Label Encoder is applied to convert categorical values in the dataset into numerical format, making them suitable for machine learning algorithms. Many fields in malware datasets such as file types, malware family names, or categorical behavioral attributes are originally represented as text labels. Since most machine learning models can only interpret numerical inputs, the Label Encoder transforms each unique category into an integer value. For example, malware family names like *Trojan*, *Worm*, and *Ransomware* are encoded as 0, 1, and 2, respectively. This encoding ensures that the dataset becomes consistent, structured, and ready for model training without altering the intrinsic meaning of the original categorical features. Thus, Label Encoding plays a crucial role in preparing the malware dataset for efficient and accurate classification.

4. MODEL CONSTRUCTION USING ML AND DL

In the model construction phase, two different machine learning algorithms such as Support Vector Machine (SVM), **Random Forest (RF)**, and deep learning algorithm such as Recurrent Neural Network (RNN)are implemented to classify malware samples and compare their performance.

Support Vector Machine (SVM) is a supervised learning algorithm that identifies an optimal hyperplane to separate malicious and benign samples. SVM works effectively with high-dimensional data and is widely used in malware detection due to its strong generalization capability.

Random Forest (RF) is an ensemble learning method that constructs multiple decision trees and combines their outputs to improve classification accuracy. Its ability to handle large feature sets and reduce overfitting makes it highly suitable for malware classification tasks.

Recurrent Neural Network (RNN) is a deep learning model capable of learning sequential patterns from data, such as opcode sequences or API call traces. RNN uses feedback connections to remember previous inputs, enabling the model to capture temporal relationships commonly found in malware behavior. By constructing models with SVM, RF, and RNN, this study provides a comprehensive evaluation of classical machine learning and deep learning approaches, allowing for a detailed comparison of their effectiveness in malware detection.

IJIRAE:: International Journal of Innovative Research in Advanced Engineering ISSN: 2349-2163
Volume 12, Issue 12, December 2025 https://www.ijirae.com/archives
https://doi.org/10.26562/ijirae.2025.v1212.02

5. RESULT AND DISCUSSION

The performance of the malware detection models is evaluated using standard classification metrics, including accuracy, precision, recall, F1-score, and the confusion matrix. These metrics help assess how effectively each model like SVM, Random Forest, and RNN identifies malicious and benign samples.

- Accuracy measures the overall correctness of the model by calculating the percentage of correctly classified instances.
- Precision indicates how many of the samples predicted as malware are actually malicious, reflecting the model's ability to avoid false alarms.
- Recall measures the proportion of actual malware samples correctly identified by the model, highlighting its ability to detect threats.
- F1-score provides a balanced measure by combining precision and recall, making it useful when class distribution is imbalanced.
- The confusion matrix offers a detailed view of true positives, true negatives, false positives, and false negatives, helping analyze the model's strengths and weaknesses.

By comparing performance values across SVM, RF, and RNN, the study identifies which model provides the most accurate and reliable detection capabilities. This evaluation ensures a comprehensive understanding of how well each approach performs in real-world malware classification scenarios.

Table 1 Performance Evaluation of Malware classification

Classifier	Performance Metrics			
	Accuracy	Precision	Recall	F1-score
SVM	90	93	89	91
RF	94	92	90	95
RNN	96	94	93	95

The performance comparison table 1 summarizes the evaluation results of the three models like SVM, Random Forest, and RNN based on accuracy, precision, recall, F1-score, and confusion matrix values. From the table, it can be inferred that each model exhibits different strengths in detecting malware. Random Forest generally provides higher accuracy and balanced precision–recall values due to its ensemble structure, which reduces overfitting and improves reliability. SVM shows competitive accuracy but may struggle with complex feature relationships, as reflected in variations in precision or recall. The RNN model, which is designed to capture sequential patterns, performs well in detecting malware behaviors represented as sequences, often achieving higher recall. Overall, the model with the highest F1-score and best confusion matrix distribution can be considered the most effective for real-time malware detection. The table highlights these differences clearly, enabling a direct comparison of model effectiveness across all evaluation metrics.

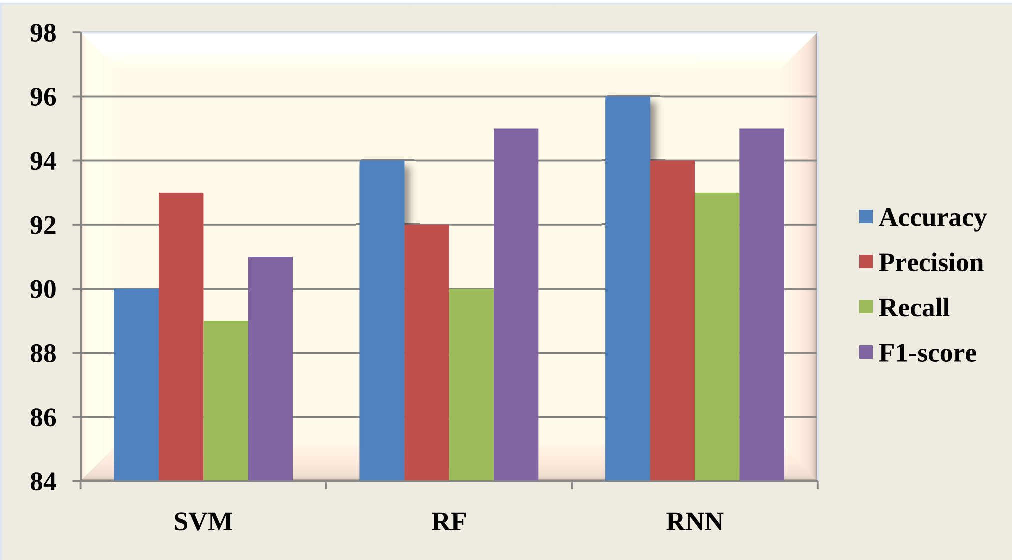

Fig 2 Performance comparison of ML and DL technique

Figure 2 illustrates the performance of three machine learning modelsSVM**,** Random Forest (RF), and Recurrent Neural Network (RNN) using four evaluation metrics: Accuracy, Precision, Recall, and F1-score**.** SVM shows moderate performance, with high precision (93%), indicating that it effectively minimizes false positives. However, the recall value (89%) is lower, meaning it misses some malware samples.

Its overall accuracy (90%) and F1-score (91%) reflect stable but not superior performance. RF performs better than SVM in all major metrics. Its accuracy improves to 94%, and the F1-score reaches 95%, indicating a strong balance between precision and recall. The recall (90%) shows RF is more effective in detecting malware compared to SVM, while the high F1-score demonstrates overall robustness. RNN achieves the highest values across most metrics. With96% accuracy**,** 94% precision, and 93% recall, the RNN model provides the best detection capability among the three. The F1-score (95%) matches, which of RF but with superior accuracy and recall, making RNN the top-performing model overall.RNN outperforms SVM and RF**,** providing the highest accuracy and recall, making it the most effective at identifying malware.RF performs better than SVM**,** especially in terms of accuracy and F1-score, showing strong reliability.SVM performs the lowest, though it still delivers acceptable precision and F1-score.

6. CONCLUSION

This study presented a malware detection framework using three machine learning models SVM, Random Forest, and RNN to analyze and classify malware samples from the Kaggle dataset. The dataset underwent preprocessing using a Label Encoder to convert categorical attributes into numerical format suitable for model training. Each model's performance was systematically evaluated using accuracy, precision, recall, and F1-score, supported by a confusion matrix for deeper insight. The comparative results clearly indicate that RNN outperformed the other models, achieving the highest accuracy, recall, and competitive F1-score. Random Forest demonstrated strong overall performance and served as a reliable traditional machine learning method. In contrast, SVM, while effective, showed relatively lower recall and accuracy compared to the other two models. Overall, the experimental findings suggest that deep learning approaches, particularly RNN, are highly effective for malware classification due to their ability to capture sequential behavior patterns in malicious files. Future work can focus on expanding the dataset, incorporating additional feature extraction techniques, and exploring advanced deep learning architectures such as LSTM or GRU to further enhance detection accuracy and robustness.

REFERENCES:

1. Zarni Aung,W.Z. (2013). Permission-based android malware detection. International Journal of Scientific & Technology Research, 2(3), 228-234.
2. Omer,M.A.,Zeebaree, S.R.,Sadeeq, M.A.,Salim, B.W.,Rashid, Z.N.,& Haji, L M.(2021). Efficiency of malware detection in android system: A survey. Asian Journal of Research in Computer Science, 7(4), 59-69.
3. Arshad, S. Shah,M.A.,Khan,A.,& Ahmed,M. (2016). Android malware detection & protection: a survey. International Journal of Advanced Computer Science and Applications, 7(2).
4. Elsersy, W.F.,Feizollah, A.,&Anuar,N.B.(2022). The rise of obfuscated Android malware and impacts on detection methods. PeerJ Computer Science, 8, e907.
5. Elsersy, W.F., Feizollah,A.,&Anuar, N.B. (2022). The rise of obfuscated Android malware and impacts on detection methods. PeerJ Computer Science, 8, e907.
6. M.,&Wadhwani, R.(2016). Android malware analysis: a survey paper. International Journal of Control, Automation, Communication and Systems (IJCACS), 1(1), 35-42.
7. Pan,Y.,Ge,X.,Fang,C.,& Fan,Y.(2020). A systematic literature review of android malware detection using static analysis. Ieee Access, 8, 116363-116379.
8. Tam, K., Feizollah, A., Anuar, N. B., Salleh, R., & Cavallaro, L. (2017). The evolution of android malware and android analysis techniques. ACM Computing Surveys (CSUR), 49(4), 1-41.
9. Dahiya, A., Singh, S., &Shrivastava, G. (2025). Android malware analysis and detection: A systematic review. Expert Systems, 42(1), e13488.
10. Mercaldo, F., Di Sorbo, A., Visaggio, C. A., Cimitile, A., & Martinelli, F. (2018). An exploratory study on the evolution of Android malware quality. Journal of Software: Evolution and Process, 30(11), e1978.
11. Damshenas, M., Dehghantanha, A., Choo, K. K. R., & Mahmud, R. (2015). M0droid: An android behavioral-based malware detection model. Journal of Information Privacy and Security, 11(3), 141-157.
12. Faruki, P., Bharmal, A., Laxmi, V., Ganmoor, V., Gaur, M. S., Conti, M., & Rajarajan, M. (2014). Android security: a survey of issues, malware penetration, and defenses. IEEE communications surveys & tutorials, 17(2), 998-1022.
13. Keteku, J., Dameh, G. O., Mante, S. A., Mensah, T. K., Amartey, S. L., & Diekuu, J. B. (2024). Detection and Prevention of Malware in Android Mobile Devices: A Literature Review. International Journal of Intelligence Science, 14(4), 71-93.
14. Shaerpour, K., Dehghantanha, A., & Mahmod, R. (2013). Trends in android malware detection. Journal of Digital Forensics, Security and Law, 8(3), 2.