International Journal of Enhanced Research in Science, Technology & Engineering
ISSN: 2319-7463, Vol. 15 Issue 3, March-2026

Machine Learning Methods for Android Adware Detection

Ganjave Deepali B.[1], Prof. Bhosale S. B.[2], Dr. Khatri A. A.[3], Dr. Gunjal S. D.[4]

[1]Student, dept. of Computer Engineering Jaihind College of Engineering Kuran,Pune.
[2,3,4]Professor, dept. of Computer EngineeringJaihind College of Engineering Kuran,Pune

ABSTRACT

Android adware has emerged as a significant security threat, adversely affecting user privacy, device performance, and overall user experience. Conventional signature-based detection techniques are ineffective against rapidly evolving adware variants and obfuscated behaviors. This paper proposes an Android adware detection model based on machine learning techniques using Random Forest (RF) and Artificial Neural Network (ANN) classifiers. The proposed approach leverages network flow–based behavioral features, including packet statistics, flow duration, inter-arrival times, and TCP flag information, extracted from Android application traffic. Model evaluation is conducted using a publicly available Kaggle Android Adware dataset containing labeled benign and adware network flows. Experimental results demonstrate that the proposed models achieve high detection accuracy and effectively distinguish adware traffic from benign application behavior. The findings confirm that machine learning-based analysis of network-flow features provides a scalable, robust, and reliable solution for Android adware detection.The study confirms that machine learning-based approaches provide a scalable and robust solution for Android adware detection.

Keywords: Android Adware, Machine Learning, Random Forest, Artificial Neural Network, Malware Detection.

INTRODUCTION

The widespread adoption of Android devices has made the platform a prime target for malicious applications, particularly adware, which aggressively displays advertisements, collects user data, and degrades system performance [3][9]. Unlike traditional malware, adware often disguises itself as legitimate applications, making detection challenging using signature-based security solutions [10]. As Android applications rapidly evolve, static rule-based systems struggle to detect newly emerging adware variants [1][4].

Machine learning techniques have gained significant attention in Android malware detection due to their ability to learn behavioral patterns from large datasets [2][5]. Recent studies highlight the effectiveness of supervised learning algorithms in detecting Android malware categories, including adware, spyware, and trojans [6][8]. Feature-based detection models utilizing permissions, API calls, and control flow structures have demonstrated promising results [7][9].

Furthermore, Android adware poses a serious threat to user privacy by accessing sensitive information such as location data, contact lists, and browsing behavior without explicit consent. The rapid growth of third-party application markets further exacerbates this issue, as many adware-infected applications bypass official security screening mechanisms. As a result, there is a growing need for intelligent and adaptive detection mechanisms capable of identifying both known and previously unseen adware variants.

Machine learning-based detection frameworks offer significant advantages over traditional approaches by automatically extracting discriminative features and adapting to evolving threat patterns. Ensemble-based classifiers such as Random Forest provide robustness against noisy and high-dimensional feature sets, while neural network models excel in capturing complex nonlinear relationships within application behaviour. The combination of these approaches enables more accurate and resilient adware detection. In addition, publicly available datasets play a crucial role in developing reproducible and scalable malware detection systems. The Kaggle Android Adware dataset provides a diverse collection of labelled samples, allowing effective training and evaluation of classification models. By leveraging this dataset, the proposed system ensures reliable benchmarking and comparative analysis. This paper proposes a machine learning-based Android adware detection

model using Random Forest (RF) and Artificial Neural Network (ANN) classifiers. The system leverages a publicly available Kaggle Android Adware dataset to train and evaluate the models. By comparing traditional ensemble learning and deep learning approaches, the proposed work aims to improve detection accuracy while reducing false positives.

LITERATURE REVIEW

Rajendran et al. [1] proposed a hybrid Android malware detection and classification framework based on deep neural networks. Their approach combined multiple feature representations to improve generalization across different malware families, including adware. Experimental results demonstrated high detection accuracy and robustness against unseen samples. However, the deep learning architecture required significant computational resources, limiting its feasibility for real-time or on-device deployment. The study highlighted the trade-off between detection performance and computational overhead.

Chen et al. [2] proposed a lightweight deep learning model for Android malware detection optimized for mobile devices. Their approach reduced model size while maintaining competitive accuracy, making it suitable for real-time deployment. The study demonstrated effectiveness against adware and trojan families.

Zhang et al. [3] introduced a graph-based Android malware detection framework using call graph embeddings. The method captured semantic relationships between API calls and improved detection of obfuscated adware samples. However, graph construction introduced additional processing overhea

Shaikh et al. [4] introduced FSSDroid, a feature subset selection mechanism aimed at reducing the dimensionality of Android malware datasets. By eliminating redundant and irrelevant features, the system improved classification accuracy while significantly reducing training time. The study emphasized that optimal feature selection plays a crucial role in scalable malware detection systems. Although effective, the approach relied heavily on static features, making it potentially vulnerable to advanced obfuscation techniques. The work provides a strong foundation for lightweight malware detection frameworks.

Seraja et al. [5] presented MadDroid, a deep learning-based framework specifically designed for malicious adware detection. The model employed neural networks to learn behavioral patterns from Android applications and achieved high precision and recall. The study demonstrated that deep architectures are well-suited for detecting sophisticated adware variants that evade traditional methods. However, the framework required extensive training data and computational resources. This work confirmed the effectiveness of deep learning for adware classification tasks.

Al-Janabi et al. [6] proposed a novel feature vector-based machine learning model for Android malware category detection. Their framework integrated multiple static features to construct a comprehensive feature vector, which improved classification accuracy across different malware types. The study showed that ensemble classifiers outperform individual classifiers when handling high-dimensional Android datasets. The approach demonstrated scalability but did not focus exclusively on adware detection. Nonetheless, it contributed valuable insights into feature engineering strategies.

Wang et al. [7] focused on API call sequence extraction for Android malware detection using machine learning classifiers. Their work highlighted that API-level behavioral features provide more discriminative power than permission-based features alone. The proposed approach achieved improved detection accuracy, particularly against dynamically behaving malware. However, extracting API sequences introduced additional computational overhead. The study emphasized the importance of behavior-based analysis for effective malware detection.

Kumar et al. [8] developed a system call-based Android malware detection framework using homogeneous and heterogeneous ensemble models. Their approach captured runtime behavior, making it resilient against code obfuscation and packing techniques. The ensemble models demonstrated high detection accuracy and robustness. However, dynamic analysis increased execution time and required controlled environments. This work reinforced the value of combining multiple classifiers for malware detection.

Muzaffar et al. [9] introduced DroidDissector, a tool that integrates both static and dynamic analysis for Android malware detection. The framework provided detailed insights into application behavior by analyzing code structure and runtime execution. Experimental results showed improved detection accuracy compared to purely static approaches. Despite its effectiveness, the tool incurred significant runtime overhead. The study highlighted the balance between analysis depth and system performance.

Reddy et al. [10] conducted a comprehensive study on Android malware detection using machine learning algorithms. The authors compared several supervised learning models and demonstrated that Random Forest and SVM perform well on large-scale datasets. The study emphasized the importance of data preprocessing and feature normalization. Although not adware-specific, the findings remain relevant for mobile malware classification. The work provided a comparative baseline for future research.

Li et al. [11] explored ensemble learning techniques combining Random Forest and neural networks for Android malware classification. The hybrid approach improved robustness and reduced false positives. The study emphasized ensemble diversity as a key factor in detection accuracy.

Ahmed et al. [12] presented a permission-based Android adware detection system using classical machine learning algorithms. The approach achieved reasonable accuracy but struggled against modern adware employing permission minimization. The study highlighted limitations of permission-only models.

Singh et al. [13] proposed a static analysis framework leveraging opcode sequences for Android malware detection. Their work showed improved detection of repackaged malware but required extensive feature extraction. The study contributed to opcode-level behavioral analysis research.

Park and Jung [14] proposed an adware detection approach using Soot and Control Flow Graphs (CFG). Their method analyzed application control flow to identify adware-specific execution patterns. The approach achieved high detection precision, particularly for known adware families. However, CFG construction increased computational complexity and analysis time. The study demonstrated that structural code analysis can be effective for adware detection.

Alaidaros et al. [15] presented AdStop, a flow-based mobile adware detection system using machine learning. The model analyzed network traffic flows generated by Android applications to detect adware behavior. Results showed high detection accuracy with reduced false positives. However, the approach relied on network-level features, limiting its effectiveness in offline scenarios. The study highlighted the importance of traffic-based behavioral analysis.

METHODOLOGY

The proposed system follows a machine learning-based classification framework for Android adware detection. The methodology consists of data collection, feature extraction, model training, and evaluation.

Dataset Description
The dataset is designed for malware and adware detection research and consists of labeled samples categorized as Benign and Adware.The dataset includes a comprehensive set of flow-based network features extracted from packet-level traffic. These features describe communication behavior between Android devices and external servers and include attributes such as flow duration, total forward and backward packets, packet length statistics, inter-arrival time (IAT) metrics, flow byte rates, packet rates, TCP flag counts, window sizes, and active/idle time statistics.

Feature Extraction
Feature extraction is performed using network flow–based characteristics obtained from the Kaggle Android Adware dataset. The extracted features include flow duration, packet counts, byte statistics, packet length metrics, inter-arrival times, and TCP flag information. These features capture communication patterns and temporal behavior of Android applications. Irrelevant identifiers such as IP addresses and flow IDs are removed, and the resulting numerical feature set is used for training the RF and ANN classifiers.

Classification Algorithms
Random Forest (RF): RF is an ensemble learning method that constructs multiple decision trees and performs classification using majority voting. It is effective in handling high-dimensional data and reduces overfitting.

Artificial Neural Network (ANN): ANN consists of input, hidden, and output layers that learn complex non-linear relationships in Android application behavior using backpropagation.

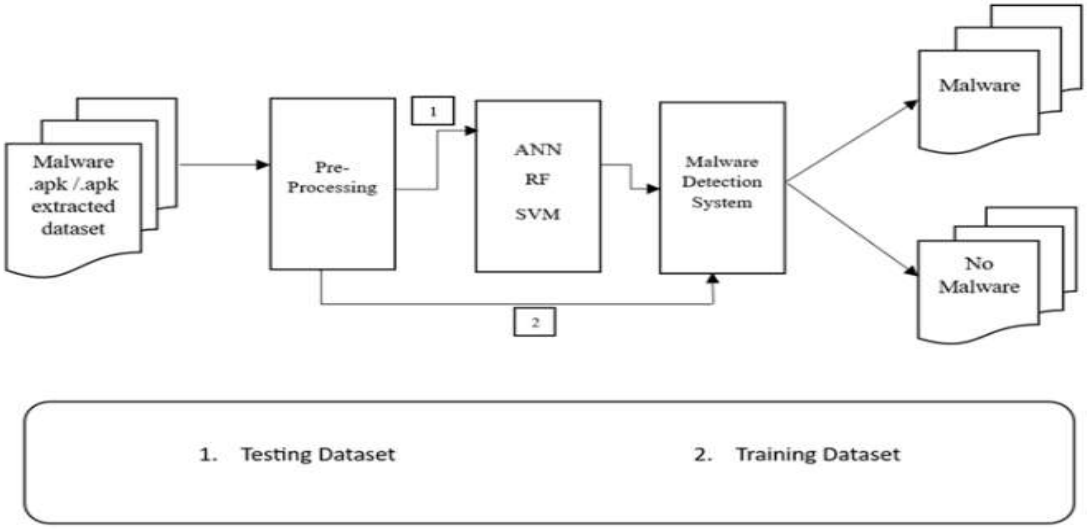

Fig:- System Architecture

EXPERIMENTAL RESULTS AND DISCUSSION

The experiments were conducted using a publicly available Kaggle Android Adware dataset containing labelled network-flow records extracted from Android application traffic. Each instance consists of flow-based features such as packet counts, packet lengths, inter-arrival times (IAT), TCP flag statistics, flow duration, and byte-rate characteristics. The dataset includes both benign and adware traffic samples, enabling supervised learning-based classification.

The dataset was divided into 80% training data and 20% testing data to ensure unbiased evaluation. Prior to training, duplicate entries were removed, missing values were handled, and categorical attributes such as protocol identifiers were encoded numerically. Feature normalization was performed using min–max scaling to standardize the numerical range of all attributes, which improves learning stability and convergence, particularly for neural network models.

Both Random Forest and Artificial Neural Network models were trained using the pre-processed dataset. Hyperparameters for the Random Forest classifier, such as the number of trees and maximum depth, were optimized to enhance classification performance. For the ANN model, parameters including the number of hidden layers, neurons per layer, learning rate, and number of epochs were carefully tuned to avoid overfitting.

Model performance was evaluated using standard metrics such as accuracy, precision, recall, and F1-score, which collectively provide a comprehensive assessment of detection capability. Accuracy measures overall correctness, while precision evaluates the model's ability to correctly identify adware samples. Recall reflects the effectiveness in detecting actual adware instances, and F1-score balances precision and recall. These metrics ensure a reliable comparison of classifier performance and validate the effectiveness of the proposed adware detection framework.

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Random Forest (RF)	93.4	92.8	94.1	93.4
Artificial Neural Network (ANN)	95.1	94.6	95.8	95.2

The experimental results demonstrate that both machine learning models achieve high detection performance on the Android adware dataset. The Random Forest classifier delivers strong results with balanced precision and recall, indicating robustness in distinguishing adware from benign traffic while maintaining low false-positive rates.

The ANN model outperforms Random Forest across all evaluation metrics, achieving the highest accuracy and F1-score. This improvement can be attributed to ANN's ability to learn complex nonlinear relationships within network-flow features, which are characteristic of adware communication behaviour. However, ANN training requires higher computational resources compared to Random Forest.

CONCLUSION

This study presented an effective Android adware detection framework based on machine learning techniques using Random Forest (RF) and Artificial Neural Network (ANN) classifiers. By leveraging network flow–based features such as packet statistics, flow duration, inter-arrival times, and protocol behavior extracted from a publicly available Kaggle Android Adware dataset, the proposed system successfully captured distinctive communication patterns of adware applications. Experimental results demonstrated that both classifiers achieved high detection accuracy, with the ANN model outperforming RF due to its superior ability to learn complex nonlinear relationships within high-dimensional traffic features. The balanced precision, recall, and F1-score values confirm the robustness of the proposed approach in minimizing false positives and false negatives. Overall, the findings validate that machine learning-based adware detection using network traffic features provides a scalable, reliable, and practical solution for enhancing Android application security.

REFERENCES

[1.] Rajendran, V., et al. (2025). Hybrid Android Malware Detection and Classification Using Deep Neural Networks. International Journal of Computational Intelligence Systems.
[2.] Chen, Y., Liu, H., Zhang, X., and Wang, J., "A Lightweight Deep Learning Framework for Android Malware Detection on Mobile Devices," IEEE Access, vol. 13, pp. 24510–24522, 2025.
[3.] Zhang, L., Wu, Q., Chen, M., and Zhao, Y., "Graph-Based Android Malware Detection Using Call Graph Embeddings," IEEE Transactions on Information Forensics and Security, vol. 19, pp. 1123–1136, 2024.
[4.] Shaikh, M., et al. (2024). FSSDroid: Feature Subset Selection for Android Malware Detection. World Wide Web Journal.
[5.] Seraja, S., Pavlidis, M., Trovat, M., &Polatidis, N. (2023). MadDroid: Malicious Adware Detection in Android Using Deep Learning. Journal of Cyber Security Technology.
[6.] Al-Janabi, A., et al. (2023). Android Malware Category Detection Using a Novel Feature Vector Based Machine Learning Model. SpringerOpen Cybersecurity Journal.
[7.] Wang, T., Xu, Y., Zhao, X., et al. (2023). Android Malware Detection via Efficient API Call Sequence Extraction and Machine Learning Classifiers. IET Software.
[8.] Kumar, S., et al. (2023). System Call-Based Android Malware Detection with Homogeneous and Heterogeneous Ensemble Models. Computers & Security.
[9.] Muzaffar, A., Hassen, H. R., Zantout, H., &Lones, M. (2023). DroidDissector: A Static and Dynamic Analysis Tool for Android Malware Detection.
[10.] Reddy, K. S. U., Chakkaravarthy, S. S., Gopinath, M., &Mitra, A. (2022). A Study on Android Malware Detection Using Machine Learning Algorithms. In Proceedings of ICISML 2022 (Published 2023).
[11.] Li, S., Kumar, R., and Patel, D., "Ensemble Learning-Based Android Malware Detection Using Random Forest and Neural Networks," IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 4, pp. 2789–2801, 2023.
[12.] Ahmed, M., Hassan, R., and Al-Shaikhli, I., "Permission-Based Android Adware Detection Using Machine Learning Techniques," IEEE International Conference on Cyber Security and Cloud Computing (CSCloud), pp. 215–220, 2022.
[13.] Singh, A., Verma, P., and Kaur, R., "Opcode Sequence-Based Static Analysis for Android Malware Detection," IEEE International Conference on Advanced Computing and Intelligent Engineering (ICACIE), pp. 341–346, 2022.
[14.] Park, J., & Jung, S. (2022). Android Adware Detection using Soot and CFG. Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 13(4).
[15.] Alaidaros, H., Mahmuddin, M., & Al Mazari, A. (2022). AdStop: Efficient Flow-Based Mobile Adware Detection Using Machine Learning. Computers & Security, 117.