



International Journal of Science, Architecture, Technology and Environment

Survey on Machine Learning-Based Android Adware Detection

Mane Pranali Rajendra^{1*}, Dr. Sanjaykumar P. Pingat²

^{1,2}STES'S Smt. Kashibai Navale College of Engineering, Pune, Computer Engineering

*Corresponding author

DOI: <https://doi.org/10.63680/ijstate1125017.015>

Abstract

The Android operating system uses a permission-based security mechanism to limit application access to device resources. However, this approach is frequently inefficient since people provide rights without careful consideration, allowing hostile entities such as malware to abuse device features. This study describes a machine learning-based method for detecting Android adware by evaluating permission requests, intents, and API usage patterns. The system is trained and validated using publicly available datasets and multiple machine learning classifiers. Key characteristics are pre-processed, including normalization and dimensionality reduction, to improve classification accuracy. The experimental results demonstrate that combining machine learning with static app metadata effectively detects adware, indicating the potential of lightweight and scalable security solutions for Android smartphones.

Keywords: Malware Detection, Feature Extraction, Machine Learning, Support Vector Machine

1. Introduction

Android's open-source nature and widespread adoption have helped it become the world's most popular mobile platform, accounting for more than 70% of the mobile operating system market. To control security, Android uses a permission-based system in which users must authorize an app access to vital services like location, internet, and contacts. Unfortunately, many users disregard permission requests and install software based on reputation or trust in the source.

With Android accounting for 70-80% of the global mobile OS market and billions of apps downloaded each year, it has become a popular target for hackers. Malware incidences increased dramatically from 120,000 to over 700,000 in a single year (2012-2013), underscoring the critical need for effective threat detection techniques. While solutions such as Google Bouncer offer some protection, sophisticated malware frequently sneaks through. This need intelligent systems capable of evaluating app behaviour and permissions in real time in order to proactively identify and mitigate threats. This technology meets the need with a machine learning-driven malware detection framework designed specifically for Android systems.

2. Section

Alaidaros et al. (2022) proposed AdStop, a machine learning-based approach for detecting mobile adware using network traffic flow features. By reducing the original 79 features to 13 using a forward-selection technique, they achieved efficient detection with a high accuracy of 98.02%. Their work demonstrates that flow-based behavioural features can successfully identify adware while reducing resource usage, though it may be limited in environments where dynamic traffic monitoring is restricted.

MadDroid, a deep learning framework that uses only static permission features to detect Android adware, was developed by Seraja et al. (2023). They outperformed other classifiers by using a Convolutional Neural Network (CNN) and a dedicated dataset of adware and benign applications. Their work highlights that even simple features like permissions can be powerful when combined with deep models, though it may fall short against adware using obfuscation or delayed execution.

A static analysis method for identifying Android adware that makes use of Control Flow Graphs (CFGs) created by the Soot framework was presented by Park and Jung in 2022. Their method, which looks at the structural patterns of malicious code, was able to detect it with an accuracy of 91.92 percent. While effective, this approach is more computationally intensive and may be vulnerable to code obfuscation or dynamic code loading techniques used by sophisticated adware.

Reddy et al. (2022) studied Android malware detection using static permissions and activity components, evaluating multiple machine learning models such as Random Forest and SVM. Their experiments demonstrated that combining component analysis with permissions results in accuracy increases of up to 95%. Despite the study's focus on general malware, its methodology lends credence to the hypothesis that feature combinations can also improve adware detection. Al-Janabi et al. (2023) proposed a multi-class Android malware classifier using a feature vector derived from system call frequency analysis and Huffman encoding. With a Random Forest accuracy of 98.7%, the system demonstrated excellent classification performance, including for adware. However, its reliance on dynamic features limits its usability in real-time on-device detection without a sandbox or emulator.

Droid, developed by Shaikh et al. (2024), focuses on feature selection for Android malware detection to maintain detection accuracy while lowering computational costs. They showed that lightweight models can still be very effective by selecting the best subsets from larger feature sets using statistical methods. While not adware-specific, their work is relevant for frameworks like yours where real-time, low-latency detection is required.

A hybrid detection system based on deep neural networks and static features like API calls, permissions, and intents was presented by Rajendran et al. (2025). Their model achieves over 98% accuracy across multiple malware families and emphasizes cross-dataset generalization. Though not adware-focused, this approach showcases the benefits of using diverse static features combined with deep learning for robust detection.

Wang et al. (2023) proposed a static analysis approach using API call sequence extraction and machine learning classifiers. By pruning irrelevant API paths and focusing on meaningful call patterns, their model improves detection rates and reduces time complexity. This method is useful for detecting malware behaviours embedded in API usage but may struggle against obfuscated or dynamically loaded code.

System call tracing and both homogeneous and heterogeneous ensemble classifiers were used by Kumar et al. (2023) to identify Android malware. Their method, which makes use of dynamic system-level features, is very accurate, especially when it comes to spotting runtime-based or evasive threats. However, like other dynamic approaches, it may not be ideal for real-time, on-device deployment without performance trade-

offs.

3. Methodology

To survey and create a machine learning-based system for detecting Android adware. It collects apps from both official and third-party sources, extracts critical aspects like permissions, API calls, and network behavior, and then preprocesses the data for analysis. Supervised machine learning models, such as Random Forest or SVM, are taught to distinguish between adware and benign programs. The models are tested for accuracy, precision, and recall, and the best-performing one is integrated into a real-time system that generates risk rankings for newly installed apps, allowing users to make more informed decisions and improving device security. The system proposes malware detection strategies. We employed I permission ranking-based feature selection, II similarity-based permission feature selection, III association rule mining, IV modified random forest classifier parameters, and V modified random forest classifier parameters. The permission ranking-based feature selection approach and the permission feature selection technique based on similarity rate the characteristics according to their frequency. The permissions are erased using the association rule mining technique, which is widely used in malware and benign programs.

4. Expected Outcome

The proposed machine learning-based framework for Android adware detection utilizes pre-extracted APK feature datasets from Kaggle to develop a lightweight, scalable, and accurate detection system. Implemented as a web-based application, the framework analyzes these static features to identify potential adware with high precision. The expected outcome is a robust, interpretable, and efficient detection model that enhances mobile security while maintaining computational simplicity. This work contributes to the advancement of intelligent, data-driven, and easily deployable web-based security solutions for Android environments.

5. Conclusion

The increase in Android-based cyber threats calls for intelligent, adaptive detection systems. This survey reviewed several techniques and proposed a robust machine learning framework that uses permission-based static features for detecting Android adware. The system combines feature selection techniques, rule-based classification, and machine learning models to achieve high detection accuracy. Future work may involve integrating lightweight dynamic features and real-time deployment models for enhanced resilience.

Acknowledgments

I would like to express my heartfelt appreciation to my guide for his constant support, insightful suggestions, and encouragement during the preparation of this survey paper. His invaluable guidance and thoughtful feedback played a significant role in the successful completion of this work.

Declaration of Conflicting Interests

The authors declare no potential conflicts of interest with respect to the research, authorship and publication of this article.

Funding

The author received no financial support for the research, authorship and publication of this article.

References

1. Alaidaros, H., Mahmuddin, M., Al Mazari, A., et al. AdStop: Efficient flow based mobile adware detection using machine learning. *Computers & Security*, Vol. 117, 2022.
2. Seraja, S., Pavlidis, M., Trovat, M., & Polatidis, N. MadDroid: Malicious Adware Detection in Android using Deep Learning. *Journal of Cyber Security Technology*, 2023.
3. Park, J., & Jung, S. Android Adware Detection using Soot and CFG. *J. Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, Vol. 13, No. 4, Dec 2022.
4. Reddy, K. S. U., Chakkaravarthy, S. S., Gopinath, M., & Mitra, A. A Study on Android Malware Detection Using Machine Learning Algorithms. *ICISML 2022 (Publ. in 2023)*.
5. Cybersecurity SpringerOpen. Android Malware Category Detection using a Novel Feature Vector based Machine Learning Model. 2023.
6. FSSDroid: Feature Subset Selection for Android Malware Detection. *World Wide Web*, 2024.
7. Hybrid Android Malware Detection and Classification Using Deep Neural Networks. *Int. J. of Computational Intelligence Systems*, 2025 (recent large scale static features + APIs + intents).
8. Wang, T., Xu, Y., Zhao, X., et al. Android malware detection via efficient API call sequences extraction and machine learning classifiers. *IET Software*, 2023.
9. A system call based Android malware detection approach with homogeneous & heterogeneous ensemble machine learning. *Computers & Security*, 2023.
10. Muzaffar, A., Hassen, H. R., Zantout, H., & Lones, M. DroidDissector: A Static and Dynamic Analysis Tool for Android Malware Detection. 2023.