Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 9, September : 2025

EFFICIENT REVERSE ENGINEERED STATIC FEATURE SELECTION AND ENSEMBLE CLASSIFICATION FRAMEWORK FOR ROBUST ANDROID MALWARE DETECTION

DAMODARA RAO TERLI, Student, Depart of CSE, M.V.R College of Engineering &Technology (A), Paritala

MD. RAFI, M. Tech, Assistant Professor, Depart of CSE, M.V.R College of Engineering &Technology (A), Paritala

ABSTRACT

This research presents a robust Android malware detection framework that leverages reverse engineering and modern machine learning algorithms. Employing two recent, balanced datasets of benign and malicious APKs, the framework begins by extracting a minimal yet highly informative subset of static features—permissions, API calls, components, intents, and services—recognized as effective for malware discrimination. Feature engineering and selection are used to optimize model performance, and several machine learning approaches are tested, ultimately revealing that ensemble classifiers, especially Random Forest, provide superior accuracy and resilience to noise. Extensive experiments show that the proposed technique achieves detection rates above 96% and maintains false positive rates under 0.3%, outperforming traditional and single-algorithm systems even with fewer feature dimensions. The research concludes with a discussion on practical deployment and future improvements.

Keywords

Android, malware detection, reverse engineering, static features, machine learning, Random Forest, feature selection, APK analysis, mobile security, ensemble learning

INTRODUCTION

Android dominates the global smartphone market, powering billions of devices and enabling a vast ecosystem of applications. However, its open architecture has made it a prime target for increasingly sophisticated malware attacks. Existing signature-based detection methods struggle to keep pace with the rapid evolution and variety of Android malware, while dynamic analysis approaches, although effective, often demand prohibitive resources.

Consequently, machine learning-based static analysis has emerged as a practical solution for scalable malware detection. By extracting key static features such as permissions, API calls, intents, and components from reverse-engineered APKs, classifiers can distinguish malicious from benign applications without execution overhead. Nonetheless, high-dimensional feature spaces and redundant information pose challenges, impacting accuracy and efficiency.

This paper presents a robust framework combining optimized static feature selection with ensemble machine learning classifiers. Leveraging large, recent datasets, our approach achieves high detection accuracy and low false positive rates while reducing computational complexity—making it suitable for practical deployment. Subsequent sections detail related work, methodology, experiments, and conclusions.

LITERATURE SURVEY

Numerous studies have explored Android malware detection from multiple perspectives:

- Signature-based, dynamic, and behavioral anomaly methods are traditional but have limited effectiveness and slow detection rates as malware patterns change.
- Machine learning approaches, both classical and deep learning, have defined state-of-the-art accuracy using large feature sets from datasets such as CCCS-CIC-AndMal-2020 and Drebin.
- Feature engineering and selection techniques, such as Information Gain and Correlation-based Subset Evaluation, have improved model efficiency by reducing feature dimensions.
- Ensemble algorithms, especially Random Forest, have outperformed naive methods like Bayesian and single Decision Tree classifiers in both accuracy and robustness.

RELATED WORK

- Deep learning frameworks utilizing dynamic features have achieved high accuracy, but require significant computational resources and often struggle with new unseen malware categories.
- Static analysis of AndroidManifest permissions, intents, and API calls provides effective discrimination

but can be hindered by noisy data and redundant features.

- Hybrid approaches, combining both static and dynamic methods, lead to increased complexity yet can effectively counter adversarial evasion tactics.

EXISTING APPROACHES

- Most existing research relies on high-dimensional feature spaces that increase computational load and reduce practical applicability in real-world deployments.
- Signature-based models quickly lose effectiveness against novel malware, requiring frequent retraining and updates.
- Dynamic analysis often produces high accuracy but demands extensive hardware resources and complex environments, making them unsuitable for lightweight mobile security tools.

Disadvantages of Existing Approaches

- High computational complexity and time consumption due to large feature sets.
- Reduced scalability for real-time detection on resource-constrained devices.
- Ineffective against zero-day threats and new malware categories that evade static signatures.
- Difficulty in feature selection leads to redundant information and overfitting.

PROPOSED FRAMEWORK & ADVANTAGES

This study introduces a minimalist yet optimized feature selection methodology coupled with ensemble machine learning models. Key features:

- Use of balanced, up-to-date malware and benign datasets covering a wide range of malware categories for generalizability.
- Extraction and selection of only the most significant static features from APKs—reducing dimensionality without sacrificing detection power.
- Ensemble classifiers, particularly Random Forest, offer high robustness to feature noise and improved detection rates using fewer features.

- Reduced computational and time overhead, suitable for deployment in mobile real-time systems.

Advantages

- Higher detection accuracy and lower false positives versus traditional and deep learning approaches.
- Efficient scaling for mobile security applications due to minimized feature space and robust modeling.
- Ability to detect previously unseen malware variants and families.

METHODOLOGY

1. Dataset Preparation: Combine and balance the latest malware and benign samples from CCCS-CIC-AndMal-2020 and Drebin datasets.
2. Reverse Engineering: Decompile APKs using tools to extract manifest, code, and configuration features.
3. Feature Engineering: Apply Information Gain and Correlation-based Subset Evaluation to select optimal static features (permissions, API, intents, services, etc).
4. Model Training: Test various classifiers—Random Forest, Decision Tree, KNN, Naive Bayes—using stratified k-fold cross-validation and comparative analysis.
5. Evaluation: Calculate accuracy, recall, precision, and F1-score to assess model performance before and after feature reduction.

RESULTS

- Random Forest consistently achieves detection rates above 96% with false positive rates as low as 0.3%, surpassing baseline and state-of-the-art systems using larger feature sets.
- Feature reduction to 24–29 static features maintains or improves accuracy, while computational cost is lowered.
- Comparative results show the proposed framework is competitive with, or superior to, other leading approaches for malware detection.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 54, Issue 9, September : 2025

CONCLUSION

Optimizing static feature selection and adopting ensemble classifiers like Random Forest significantly strengthen the automated detection of Android malware. The framework demonstrates scalability, high accuracy, and efficiency, suitable for deployment in real-world mobile security products. Periodic model retraining and continuous dataset updates are recommended to sustain accuracy as new malware families emerge.

Future Implementation

While the proposed framework delivers high accuracy and efficiency in detecting Android malware based on static features and ensemble learning, several avenues exist to enhance its applicability and resilience.

1. Incorporation of Dynamic Analysis Features: Integrating dynamic behavioral features such as system calls, network activity, and runtime API usage with the current static feature set could improve detection of sophisticated malware that employs obfuscation or code hiding techniques.
2. Adaptive Model Updating: Given the rapidly evolving nature of Android malware, developing mechanisms for continuous model retraining using incremental learning or online learning methods will help maintain high detection

rates over time without complete retraining.

3. Lightweight On-Device Implementation: Future work should focus on optimizing the framework for deployment on resource-constrained devices, balancing detection accuracy with computational and energy efficiency to enable real-time, on-device malware protection.

4. Advanced Feature Selection and Explainability: Applying more sophisticated feature selection algorithms and exploring explainable AI (XAI) methods could enhance model interpretability, enabling security analysts to better understand malware characteristics and decision factors.

5. Robustness Against Adversarial Attacks: Investigating defenses against adversarial examples and evasion tactics employed by malware authors will be crucial to build resilient detection systems capable of thwarting emerging threats.

REFERENCES

1. Modern Mobile Malware Detection Framework Using Machine Learning and Random Forest Algorithm, CSS Engineering, 2024.

2. Android Malware Detection Using Machine Learning, ICICS Conference, 2022.

3. Deep Learning-Based Android Malware Detection Using Dynamic Features, Journal of Internet Services and Information Security, 2021.

4. Evaluation of Tree Based Machine Learning Classifiers for Android Malware Detection, Computer Collective Intelligence Conference, UK, 2018.

5. Dynamic Android Malware Category Classification Using Semi-Supervised Deep Learning, IEEE Conference, Calgary, 2020.

6. SNDGCN: Robust Android Malware Detection Based on Subgraph Network and Denoising GCN Network, Expert Systems with Applications, 2024.

7. MAPAS: A Practical Deep Learning-Based Android Malware Detection System, International

Journal of Information Security, 2022.