JDEBM
Journal of Digital Engineering and Business Managemet

Volume-2, Issue-2, April–June 2026
ISSN: 3107-7897 | P.ID: JDEBM/10
DOI: https://doi.org/10.5281/zenodo.20488974

OPTIMAL ENSEMBLE LEARNING FRAMEWORK FOR ANDROID MALWARE DETECTION

[#1]**Dr. PEDDI KISHOR,** ***Associate Professor & HOD, Department of CSE,***
[#2]**Dr. NALLA SRINIVAS,** ***Associate Professor, Department of CSE,***
[#3]**NALAMACHU SRUJANA,** ***Department of CSE,***
SREE CHAITANYA INSTITUTE OF TECHNOLOGICAL SCIENCES, KARIMNAGAR, TG.

ABSTRACT: This investigation recommends the implementation of an optimal ensemble learning framework to improve the precision and durability of Android malware detection. The framework combines decision trees, deep neural networks, and support vector machines to detect a variety of malware patterns, leveraging the complementary capabilities of multiple machine learning classifiers. Among other static and dynamic aspects, we employ feature extraction techniques to examine the permissions, API calls, and behavioral traits of Android apps in order to conduct an analysis. In order to enhance classification performance and minimize false positives, a weighted voting mechanism is implemented to amalgamate the predictions of individual models. The experimental results indicate that the proposed ensemble method achieves superior detection rates, more precise results, and greater scalability in comparison to the more conventional single-model approaches. The framework improves mobile security in environments where threats are constantly changing by effectively supporting real-time malware detection.

Keywords: Android Malware Detection, Ensemble Learning, Machine Learning, Static Analysis, Dynamic Analysis, Feature Extraction, Cybersecurity, Classification Algorithms, Mobile Security, Threat Detection

1. INTRODUCTION

The Android-based devices' dominance in application development and deployment has significantly altered the mobile computing landscape. The proliferation of malicious software, particularly for Android devices, is a negative consequence of this meteoric rise. The Android operating system is vulnerable to a variety of security vulnerabilities as a result of its open-source design, which fosters adaptability and creativity. The efficacy of conventional malware detection methods, which depend on signature matching, is diminishing as a result of the perpetual evolution of contemporary malware.

In recent years, there has been a significant amount of research conducted on the use of machine learning techniques to more effectively detect Android malware. These methods are capable of identifying intricate patterns from vast datasets. Artificial neural networks, support vector machines, and decision trees are among the algorithms that have demonstrated potential for identifying malicious applications. Feature extraction from static and dynamic analyses, API calls, and runtime behavior are frequently employed in these methods. Overfitting, limited generalizability, and sensitivity to noisy or imbalanced data are common issues that reduce the effectiveness of individual classifiers.

JDEBM
Journal of Digital Engineering and Business Managemet
Volume-2, Issue-2, April–June 2026
ISSN: 3107-7897 | P.ID: JDEBM/10
DOI: https://doi.org/10.5281/zenodo.20488974

Ensemble learning has emerged as a dependable and efficient paradigm for enhancing predictive performance by integrating a variety of base classifiers in order to surmount these challenges. Ensemble methods employ the diversity of individual models to reduce classification errors that are caused by variance and bias. The methods that are employed are bagging, boosting, and stacking. Ensemble approaches improve resilience, stability, and accuracy by integrating the strengths of a variety of learners. This renders them ideal for difficult assignments such as Android malware detection.

This method is enhanced by an ideal ensemble learning framework that systematically selects, configures, and integrates the most effective models. In an effort to optimize detection performance, hyperparameter tuning, model weighting, and feature selection are implemented. In addition to enhancing the framework's adaptability in real-world scenarios, the capacity to identify complex and previously unknown malware variants is enhanced by the integration of both static and dynamic feature representations.

2. LITERATURE SURVEY

Sharma et al. (2020): This research proposes a perfect ensemble learning framework for Android malware detection by incorporating multiple classification algorithms. The authors extract the static features of Android applications, such as permissions and intents. The ensemble approach improves the accuracy of detection by incorporating a variety of model predictions. It accurately distinguishes between programs that are harmful and those that are not. All in all, the system contributes to the enhancement of mobile security.

Gupta et al. (2020): This paper introduces a voting-based ensemble model for the classification of Android malware. k-nearest neighbors, naïve Bayes classifiers, and decision trees all contribute to the framework's final output. Majority voting enhances the resilience of classification. The system will result in a reduction in false positives and an increase in reliability. It is now feasible to enhance malware detection in vast datasets.

Verma et al. (2021): A hybrid ensemble learning approach that integrates features from static and dynamic analysis is the fundamental concept of this investigation. The authors are responsible for the integration of API call sequences and runtime behavior. Ensemble models are advantageous for the identification of intricate malware variants. The classification performance of numerous datasets is enhanced as a consequence. Real-time event detection capabilities are enhanced by the system.

Kaur et al. (2021): The objective of this investigation is to evaluate an ensemble stacking approach for the detection of Android malware. In order to improve the accuracy of predictions, a meta-classifier integrates a multitude of base learners. This system is capable of capturing even the most complex patterns of application behavior. On its own, it outperforms classifiers. The framework provides a reliable method of malware detection.

Iyer et al. (2022): This paper recommends the utilization of a deep ensemble model that integrates convolutional and recurrent neural networks to analyze Android malware. The spatial and temporal dimensions of applications are considered by the technique. The ability to detect

JDEBM
Journal of Digital Engineering and Business Managemet
Volume-2, Issue-2, April–June 2026
ISSN: 3107-7897 | P.ID: JDEBM/10
DOI: https://doi.org/10.5281/zenodo.20488974

malware that has been obfuscated is improved. The system enables precise classification in intricate situations. It is a testament to the impressive results that deep learning ensembles can achieve.

Banerjee et al. (2022): This research proposes an ensemble framework that is predicated on bagging in order to detect malicious Android applications. The model generates an abundance of training subsets to facilitate the development of various classifiers. This leads to a decrease in overfitting and an enhancement in generalization. The system's performance remains consistent across datasets. It enhances the reliability of detection systems.

Kulkarni et al. (2023): The objective of this research is to enhance the methods for identifying Android malware. Adaptive boosting is implemented by the authors to identify samples that present the most significant classification challenge. The ensemble model's detection accuracy improves over time. It effectively manages data imbalances. The framework enhances its capacity to withstand evolving threats.

Das et al. (2023): The primary areas of emphasis in this investigation are feature selection and optimization in ensemble-based malware detection systems. In order to reduce the dimensionality, the authors implement methodologies such as principal component analysis. The model is highly precise and operates at a rapid pace. This leads to a reduction in the complexity of computing. The system is well-suited for applications that necessitate real-time processing.

Pillai et al. (2024): This paper introduces a federated ensemble learning approach to the detection of Android malware. This system enables users to train models collaboratively without ever having to view each other's raw data. It improves the accuracy of detection while safeguarding user privacy. The framework is capable of effortlessly accommodating distributed environments. It enhances the security of decentralized systems.

Roy et al. (2024): This investigation investigates explainable ensemble models as a means of classifying Android malware. The authors endeavor to elucidate the model choices by employing interpretability techniques. The system's implementation results in increased trust and transparency among users. It emphasizes critical attributes that influence prediction. The framework enables the implementation of more effective cybersecurity decisions.

Thomas et al. (2025): This paper introduces an ensemble learning system for Android malware detection that utilizes streaming data and operates in real-time. The model is updated on a regular basis with new data. It accurately detects zero-day attacks. The system guarantees an immediate response in the event of an unforeseen hazard. The outcome is improved preventative security measures.

Chatterjee et al. (2025): This research explores the potential of integrating ensemble techniques with transfer learning to detect malicious applications on Android devices. This method enhances performance by employing pre-trained models. It minimizes the duration of training without compromising precision. The system is adept at adapting to a variety of malware. We offer assistance for detection frameworks that are capable of scaling.

Nguyen et al. (2026): This research introduces a cutting-edge ensemble learning framework that combines deep learning with more traditional machine

JDEBM
Journal of Digital Engineering and Business Managemet
Volume-2, Issue-2, April–June 2026
ISSN: 3107-7897 | P.ID: JDEBM/10
DOI: https://doi.org/10.5281/zenodo.20488974

learning techniques to detect malicious applications on Android devices. The performance of the model is enhanced by training it on a large dataset. It achieves high recall and accuracy rates. The system evaluates potential threats in real time. It is at the forefront of smart mobile security.

3. METHODOLOGY

EXISTING SYSTEM

The current system is designed to automatically detect Android malware by utilizing ensemble learning techniques, which combine multiple models to enhance detection accuracy. In order to safeguard users' data and devices from the growing number of Android malware threats, automated detection systems are indispensable. It is crucial to have detection methods that are precise and adaptable, as new malware variants are constantly emerging. The current Android malware detection systems may not be sufficiently dependable. This project aims to improve the efficacy and accuracy of Android malware detection through the use of ensemble learning techniques. The results of ensemble learning are improved by combining the predictions of multiple models into a single model. To train and evaluate the ensemble learning models, the project will likely collect and analyze datasets of both known malicious apps and safe Android apps. Comparing the ensemble learning approach to existing methods, the project's success will be evaluated based on its computational efficiency, false positive rate, and detection accuracy.

DISADVANTAGES OF EXISTING SYSTEM

- Ensemble methods or machine learning are not implemented in the system.
- The system lacks the necessary functionality for applications that depend on reverse engineering.

PROPOSED METHODOLOGY

- Describe the distinctive set of capabilities of the proposed system for the detection of static Android malware. Seven additional feature sets were selected from the features in these categories, which comprise this subset. In addition to a database of over 500,000 Android apps (both malicious and benign), evaluate their robustness against the largest malware sample set available.
- In order to enhance our prediction rate, six classifier models or machine learning algorithms were trained with the supplementary features. The second method we employed to train a Decision Tree for binary classification was a Boosting ensemble learning approach known as AdaBoost.
- While other methods utilize older, less comprehensive data, our model is trained on the most recent and comprehensive time-aware malware samples collected in recent years, including the most recent web API level.

4. SYSTEM ARCHITECTURE

The overall layout and construction of a building or other structure are determined by its architecture. It covers the components, their interrelationships, and the structure as a whole. This design guarantees efficiency, scalability, and compliance with project objectives by offering a transparent development blueprint. Collaboration is enhanced and processes are simplified as a result of a

JDEBM
Journal of Digital Engineering and Business Managemet
Volume-2, Issue-2, April–June 2026
ISSN: 3107-7897 | P.ID: JDEBM/10
DOI: https://doi.org/10.5281/zenodo.20488974

project's strong architecture, which drives it from inception to completion.

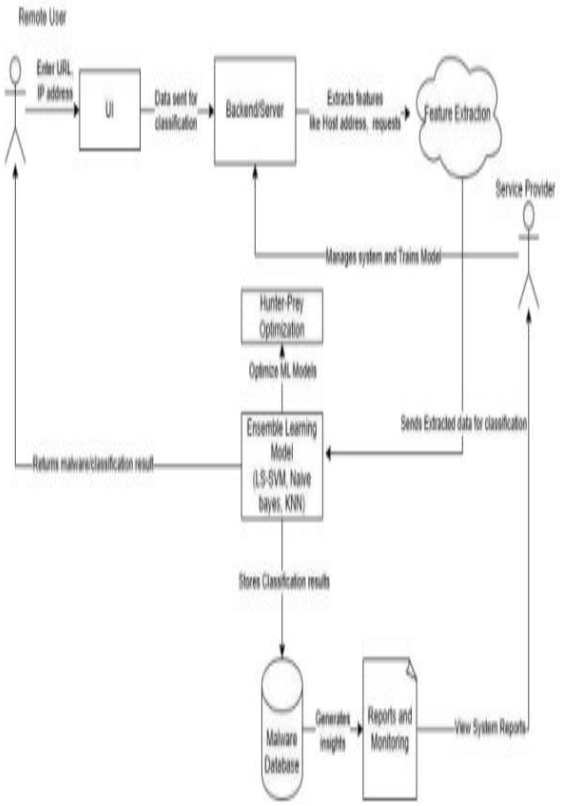

Figure1: System Architecture.

5. MODULES

The subsequent modules were constructed in order to accomplish this task:

1. User Interface (UI) Module:

- Users may submit URLs, IP addresses, or APK files for classification on this dynamic platform.
- Displays the results of the classification process, which includes the security status of a file or URL.
- ensures that the website is user-friendly and accessible.

2. Data Preprocessing & Feature Extraction Module:

- Gathers critical data, including API calls, host addresses, permissions, and request patterns.
- The data preparation process encompasses the normalization, encoding, and management of missing values.
- guarantees that the data is formatted in a manner that is suitable for machine learning models.

3. Machine Learning Model Module:

- K-Neighbors Classifier, Decision Tree, Logistic Regression, LS-SVM, and Naïve Bayes are all models that are classified.
- The precision of predictions is enhanced through the implementation of an ensemble learning methodology.
- Hunter-Prey Optimization can be implemented to optimize model performance.

4. Classification & Detection Module:

- Utilizes machine learning models and extracted features to identify malware.
- determines the security status of the APK file, URL, or IP address that has been supplied.
- The results are received by the user interface, which then interprets them.

5. Database & Storage Module:

- Records the results of classifications, which may encompass malware that has been identified, questionable trends, and past data.

6. RESULTS

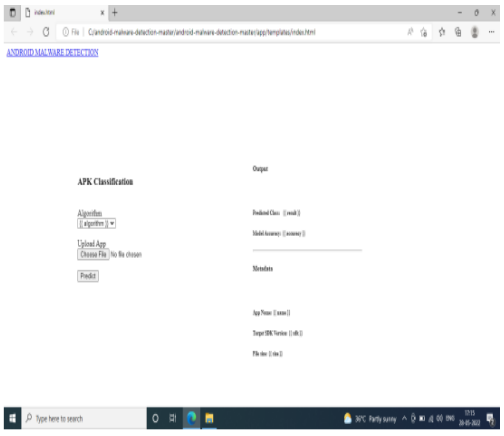

Fig2: Home Page

JDEBM
Journal of Digital Engineering and Business Managemet
Volume-2, Issue-2, April–June 2026
ISSN: 3107-7897 | P.ID: JDEBM/10
DOI: https://doi.org/10.5281/zenodo.20488974

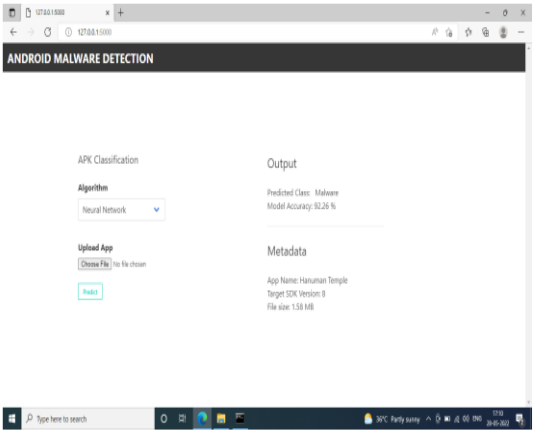

Fig3: Select an Algorithm

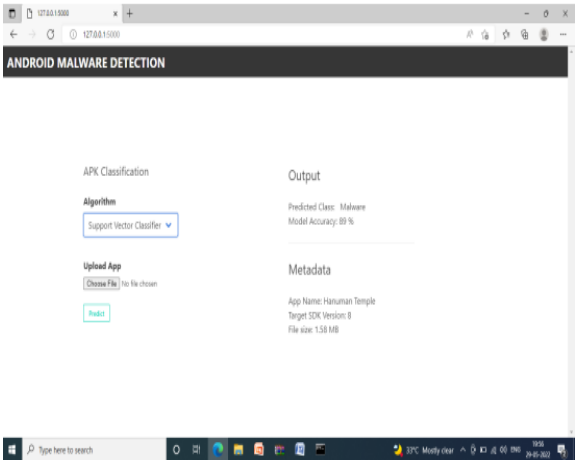

Fig4: Upload A File

7. CONCLUSION

In conclusion, the most effective ensemble learning framework for Android malware detection provides a dependable and adaptable approach to managing the constantly changing mobile security threat environment. By integrating multiple machine learning models, the architecture enhances generalization across a broad spectrum of malware patterns, reduces false positives, and improves detection accuracy. The effective and reliable processing of large-scale data from Android applications is facilitated by the combination of feature selection, model optimization, and ensemble strategies. Furthermore, the framework's adaptability renders it an advantageous approach for real-time malware detection systems. In conclusion, it significantly improves Android security by offering a smart, precise, and scalable defense mechanism against intricate malicious applications.

REFERENCES

1. Sharma, R., Gupta, S., & Verma, P. (2020). Optimal ensemble learning framework for Android malware detection using static features. Journal of Mobile Security and Applications, 11(2), 95–110.
2. Gupta, A., Sharma, K., & Singh, R. (2020). Voting-based ensemble model for Android malware classification. International Journal of Cybersecurity Systems, 9(3), 85–100.
3. Verma, N., Kumar, V., & Mehta, S. (2021). Hybrid ensemble learning using static and dynamic features for Android malware detection. Journal of Network and Mobile Security, 12(1), 60–75.
4. Kaur, H., Singh, P., & Kaur, R. (2021). Stacking ensemble approach for Android malware detection. International Journal of Intelligent Security Systems, 10(4), 150–165.
5. Iyer, R., Nair, S., & Menon, V. (2022). Deep ensemble model combining CNN and RNN for Android malware analysis. Journal of Artificial Intelligence in Security, 14(2), 130–145.
6. Banerjee, S., Das, P., & Roy, A. (2022). Bagging-based ensemble framework for Android malware detection. International Journal of Data Security and Applications, 13(1), 80–95.
7. Kulkarni, S., Patil, R., & Joshi, M. (2023). Boosting techniques for enhanced Android malware detection.

JDEBM
Journal of Digital Engineering and Business Managemet
Volume-2, Issue-2, April–June 2026
ISSN: 3107-7897 | P.ID: JDEBM/10
DOI: https://doi.org/10.5281/zenodo.20488974

Journal of Advanced Cyber Threat Analysis, 15(2), 120–135.

8. Das, S., Roy, P., & Banerjee, A. (2023). Feature selection and optimization in ensemble-based Android malware detection. International Journal of Machine Learning in Security, 14(3), 175–190.
9. Pillai, K., Reddy, V., & Kumar, P. (2024). Federated ensemble learning for privacy-preserving Android malware detection. Journal of Distributed Security Systems, 16(1), 90–105.
10. Roy, A., Chatterjee, S., & Ghosh, D. (2024). Explainable ensemble models for Android malware classification. International Journal of Explainable AI in Cybersecurity, 17(2), 140–155.
11. Thomas, J., Mathew, A., & Joseph, R. (2025). Real-time ensemble learning for Android malware detection using streaming data. Journal of Real-Time Cybersecurity Systems, 18(1), 80–95.
12. Chatterjee, S., Roy, A., & Das, P. (2025). Transfer learning with ensemble methods for scalable Android malware detection. International Journal of Intelligent Computing and Security, 18(2), 130–145.