Journal of Future Artificial Intelligence and Technologies

E-ISSN: 3048-3719

Research Article

An Optimized Stacking Ensemble of Deep Learning and Machine Learning Models with Boruta-Based Feature Selection for Android Malware Detection

Maiauduga Abdullahi Danladi [1], Maryam Sufiyanu Masari [1], Ubakaghinwa Paul Chigbu [2], and Abdulrashid Abdulrauf [3,*]

[1] Department of Cybersecurity, Faculty of Computing, Air Force Institute of Technology, Kaduna 800001, Kaduna State, Nigeria; e-mail : abdullahi.maiauduga@afit.edu.ng; msufiyanu90@gmail.com
[2] Department of Information and Communication Technology, National Office for Technology Acquisition and Promotion, Abuja 900281, FCT Abuja, Nigeria; e-mail : chigbuuba@gmail.com
[3] Department of Computer Science, Federal Polytechnic Kaltungo, Kaltungo 770101, Gombe State. Nigeria; e-mail : abdulrashid.abdulrauf2022@nda.edu.ng
* Corresponding Author : Abdulrashid Abdulrauf

Abstract: Malware, particularly within the Android ecosystem, continues to evolve in complexity, rendering traditional signature-based detection techniques ineffective against new and obfuscated threats. This study addresses the challenge of achieving highly accurate and robust malware detection in the presence of high-dimensional and potentially noisy feature spaces. The objective is to develop a feature-optimized hybrid framework that leverages the complementary strengths of deep learning and machine learning models. To this end, a stacking-based hybrid model integrating TabNet, Deep Neural Network, and XGBoost as base learners, with Logistic Regression as the meta-learner, is proposed. The methodology incorporates Boruta-based feature selection to eliminate irrelevant and redundant features, thereby improving model generalization, efficiency, and robustness. Rather than relying on explicit data resampling or class balancing techniques, the proposed approach enhances detection stability under naturally imbalanced data distributions through feature optimization and heterogeneous ensemble learning. The model is evaluated on two benchmark datasets, TUANDROMD and CIC-AndMal2017, to ensure robustness and generalizability. Experimental results demonstrate that on the CIC-AndMal2017 dataset, the hybrid model achieved an accuracy of 0.978945, precision of 0.969766, recall of 0.970508, F1-score of 0.970137, and AUROC of 0.996030, outperforming all individual base models. Similarly, on the TUANDROMD dataset, the hybrid framework attained superior performance with an accuracy of 0.997760, precision of 0.997203, recall of 1.000000, F1-score of 0.998599, and AUROC of 0.999423, again surpassing the base learners. These results confirm that the proposed feature-optimized ensemble stacking approach improves malware detection performance and provides a robust and scalable solution for real-world cybersecurity applications.

Keywords: Android Malware; Boruta; Class Weighting; Deep Learning; Ensemble Learning; Feature Selection; Machine Learning; Malware Detection.

Received: March, 28th 2026
Revised: April, 28th 2026
Accepted: April, 29th 2026
Published: April, 30th 2026
Curr. Ver.: April, 30th 2026

Copyright: © 2026 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY SA) license (https://creativecommons.org/licenses/by-sa/4.0/)

1. Introduction

In recent years, the rapid expansion of Internet of Things (IoT) technologies and the widespread adoption of smartphones have significantly transformed modern digital ecosystems. Among mobile operating systems, Android has emerged as the dominant platform globally due to its open-source architecture, flexibility, and affordability. However, this rapid growth has also expanded the attack surface for cyber threats, making Android devices a primary target for malicious activities [1], [2]. As mobile devices increasingly handle sensitive

personal, financial, and organizational data, ensuring their security has become a critical concern.

One of the most severe and rapidly evolving threats within the Android ecosystem is ransomware. Android ransomware infiltrates devices, encrypts critical user data, and demands payment for its release, leading to substantial financial losses, data breaches, and operational disruptions [3], [4]. The growing sophistication of ransomware, combined with its ability to exploit system vulnerabilities and user behavior, has amplified its impact on both individuals and organizations. Furthermore, Android's fragmented ecosystem and the widespread use of third-party application stores often lacking rigorous security controls further increase the risk of malware infiltration [5].

Detecting Android ransomware remains a complex challenge due to the evolving nature of modern malware. Advanced evasion techniques such as obfuscation, polymorphism, and encryption enable malware to bypass traditional security mechanisms [6], [7]. Additionally, the diversity of malware behaviors, coupled with the resource constraints of mobile devices, limits the effectiveness of conventional detection approaches. The dynamic nature of application behavior and network traffic further complicates real-time detection, often resulting in missed threats or elevated false-positive rates [8]. These challenges necessitate the development of more adaptive, robust, and intelligent detection frameworks.

Traditional malware detection techniques, including signature-based and anomaly-based methods, have demonstrated limitations in addressing modern threats. Signature-based approaches depend on known patterns and are ineffective against zero-day attacks, while anomaly-based methods, although capable of identifying unknown threats, often suffer from high false alarm rates [9]. Consequently, machine learning (ML) and deep learning (DL) techniques have been increasingly adopted. ML models such as Decision Trees, Random Forests, and Support Vector Machines provide efficient and interpretable detection, whereas DL models enable automatic feature extraction and improved modeling of complex patterns [10], [11]. Despite these advantages, standalone ML and DL models often struggle with generalization and robustness, particularly when confronted with heterogeneous and potentially imbalanced datasets [12]. To address these limitations, recent studies have explored hybrid and ensemble learning strategies that integrate multiple models to leverage complementary strengths. Such approaches aim to improve detection accuracy, robustness, and generalization across diverse datasets [13]. Nevertheless, existing methods often focus on either feature selection or model architecture in isolation, leaving a gap for integrated frameworks that jointly optimize both aspects.

In this study, a stacking-based hybrid framework is proposed to enhance Android ransomware detection. The proposed model integrates TabNet, Deep Neural Network (DNN), and Extreme Gradient Boosting (XGBoost) as base learners, with Logistic Regression serving as the meta-learner. Boruta-based feature selection is employed to identify relevant features and eliminate redundancy, thereby improving model efficiency and generalization. The framework is evaluated on two benchmark datasets, CIC-AndMal2017 and TUANDROMD, to ensure robustness and generalizability across different data distributions.

The selection of TabNet, XGBoost, and DNN is driven by their complementary learning characteristics. TabNet is specifically designed for tabular data and incorporates sequential attention-based feature selection, enabling interpretability and efficient feature utilization. XGBoost offers strong generalization through gradient boosting and is particularly effective for structured data. DNN, in contrast, are capable of capturing complex nonlinear relationships. By integrating these heterogeneous learners within a stacking architecture, the proposed framework aims to reduce model bias and variance while enhancing overall generalization performance.

The primary contribution of this study lies in the unified integration of feature-level optimization and model-level learning. By combining Boruta-based feature selection with a heterogeneous stacking ensemble, the proposed framework simultaneously addresses feature redundancy and model robustness. This integration enables improved detection performance and a more balanced precision–recall trade-off compared to standalone approaches. Importantly, rather than relying on explicit resampling or synthetic class balancing techniques, the proposed framework improves robustness under imbalanced data distributions through feature optimization, class-aware feature evaluation, and ensemble learning.

To formally guide this work, the following research hypothesis is proposed: the integration of Boruta-based feature selection with a heterogeneous stacking framework yields

superior and more stable performance in Android malware detection compared to standalone ML and DL models. This improvement is expected to manifest not only in overall accuracy but also in enhanced recall, reduced false negative rates, and a more balanced precision–recall trade-off, thereby increasing suitability for real-world security applications. The remainder of this paper is organized as follows. Section 2 reviews related work, Section 3 describes the proposed methodology, Section 4 presents experimental results and discussion, and Section 5 concludes the paper with key findings and future research directions.

2. Related Work

Recent advancements in Android malware detection have evolved along three major research directions: feature optimization, class imbalance handling, and hybrid or ensemble learning approaches, alongside emerging interest in federated and privacy-preserving frameworks. While these directions have significantly improved detection accuracy, their relative strengths, limitations, and interactions reveal important gaps that continue to motivate further research.

Feature optimization techniques have been widely adopted to enhance detection performance by reducing dimensionality and improving feature relevance. Studies such as Jilani and Samreen [14] and Ibadullah et al. [15] demonstrate that combining feature extraction and selection methods, including Singular Value Decomposition, ReliefF, and optimization algorithms such as the Whale Optimization Algorithm, can yield accuracies exceeding 99%. Similarly, Pristyanto et al. [16] proposed a hybrid filtering and wrapper-based feature selection framework, achieving improved computational efficiency and robustness across multiple datasets. Despite these promising results, such approaches often assume relatively clean and balanced data distributions. As highlighted by Etem [17], excessive dimensionality reduction may lead to the loss of discriminative information, introducing a trade-off between model simplicity and predictive performance. Moreover, these methods are rarely evaluated in conjunction with other practical challenges, particularly class imbalance and model generalization, which limits their applicability in real-world scenarios.

Class imbalance remains a well-recognized challenge in Android malware detection, where benign applications typically outnumber malicious samples. To address this issue, resampling techniques such as SMOTE and hybrid strategies like SMOTE-ENN have been widely explored. Fenjan et al. [18] demonstrated that such methods can significantly improve minority class detection, particularly in terms of recall. However, while resampling techniques improve class distribution, they may introduce synthetic noise and increase the risk of overfitting, especially when not combined with effective feature selection. Masari et al. [13] further demonstrated that combining Random Forest with SMOTE-Tomek resampling yields strong performance, making it a relevant benchmark for imbalance-aware detection systems. Nevertheless, reliance on resampling alone does not fully address feature redundancy and model generalization challenges. In contrast to these approaches, the present study does not employ explicit class balancing techniques such as SMOTE, undersampling, or oversampling. Instead, it focuses on Boruta-based feature optimization and heterogeneous ensemble learning to improve robustness under naturally imbalanced data distributions. In this context, class weighting is applied within the Boruta Random Forest to support fair feature importance estimation and should not be interpreted as a data-level balancing strategy.

In parallel, hybrid and ensemble learning approaches have gained prominence due to their ability to leverage the strengths of multiple algorithms. Comparative studies by Al Ali et al. [19] and Akkaya and Altay [20] show that traditional ML models, including Support Vector Machines, Sequential Minimal Optimization, and logistic classifiers, can achieve high accuracy when properly tuned. However, ensemble methods such as Random Forest, XGBoost, LightGBM, and stacking techniques consistently outperform standalone models, as reported by Abdulla [21] and Shah [22], with accuracies exceeding 99%. These improvements are largely attributed to the ability of ensemble models to reduce variance and capture diverse decision boundaries. Despite these advantages, ensemble approaches often require careful parameter tuning and may introduce additional computational complexity.

Deep learning and hybrid architectures further extend detection capabilities by modeling complex nonlinear relationships in large-scale datasets. Ambekar et al. [23] proposed TabLSTMNet, integrating TabNet and Long Short-Term Memory (LSTM) networks to achieve high predictive performance while incorporating explainability through LIME and

SHAP. Similarly, Mehmood et al. [24] combined LSTM with fuzzy logic to address uncertainty in real-time detection scenarios. Patel et al. [25] proposed a sequential DL approach based on API call sequences, comparing LSTM and Transformer architectures under controlled experimental settings. Their findings indicate that Transformer-based models outperform LSTM, achieving 95.54% accuracy with improved precision, recall, and F1-score, highlighting the effectiveness of attention mechanisms in capturing behavioral dependencies. Other approaches, including Trandroid [26] and H2O AutoML frameworks [27], also report strong performance exceeding 99%. However, DL models are often computationally expensive, less interpretable, and sensitive to noisy or redundant features. Their performance remains highly dependent on input feature quality, reinforcing the importance of effective feature selection.

Emerging research has also explored federated and privacy-preserving learning frameworks to address data security concerns in distributed environments. Abdellah et al. [[28] introduced CNN-Fed, which enables collaborative model training without centralized data sharing, demonstrating strong performance across multiple benchmark datasets. Similarly, Daulay et al. [29] proposed FedGCL, integrating graph neural networks and contrastive learning to achieve high accuracy while preserving privacy. Additional studies [30], [31] further confirm the feasibility of federated approaches in distributed malware detection. However, these frameworks introduce challenges related to communication overhead, model convergence, and data heterogeneity across clients, which may affect overall performance and scalability.

A critical observation across the literature is that most studies tend to focus on isolated improvements either feature optimization, imbalance handling, or model enhancement rather than integrating these components into a unified framework. Feature selection methods often overlook class distribution effects, while imbalance handling techniques frequently neglect feature redundancy and model interaction. Similarly, hybrid and ensemble approaches, although achieving high accuracy, do not consistently address feature quality or dataset variability. Furthermore, many studies rely on single-dataset evaluations, limiting their ability to demonstrate generalization across diverse data distributions.

Motivated by these gaps, this study proposes a feature-optimized hybrid stacking framework that integrates TabNet, XGBoost, and DNN as base learners, with Logistic Regression as a meta-learner. The framework incorporates Boruta-based feature selection to reduce redundancy and enhance feature relevance, while leveraging heterogeneous ensemble learning to improve robustness and generalization. Importantly, rather than relying on explicit resampling or synthetic balancing techniques, the proposed approach aims to improve detection performance under imbalanced data distributions through feature optimization, class-aware feature evaluation, and model diversity. In addition, the model is evaluated on multiple benchmark datasets, including CIC-AndMal2017 and TUANDROMD, to ensure robustness and cross-dataset generalizability. By jointly addressing feature quality, model diversity, and evaluation robustness within a single framework, this work seeks to provide a more comprehensive and practically applicable solution for Android malware detection.

3. Proposed Method

This section describes the datasets, data preprocessing procedures, feature selection strategy, and model architecture employed in this study. The overall workflow of the proposed framework is illustrated in Figure 1, which presents a structured view of the data flow, from preprocessing and feature selection to model training and evaluation.

3.1. Dataset

This study utilizes two benchmark datasets, namely TUANDROMD and CIC-AndMal2017, obtained from publicly available sources. These datasets are widely used in Android malware detection research and provide diverse characteristics for evaluating model robustness and generalizability.

3.1.1. TUANDROMD Dataset

The TUANDROMD dataset used in this study is a preprocessed version obtained from the UCI Machine Learning Repository [32]. It consists of 4,465 Android application samples, each described by 241 integer-valued features. The dataset is structured for binary classification, where each instance represents either a malware or a benign (goodware) application. The

class labels are encoded as 1 for malware and 0 for goodware. A statistical overview of the dataset is presented in Table 1.

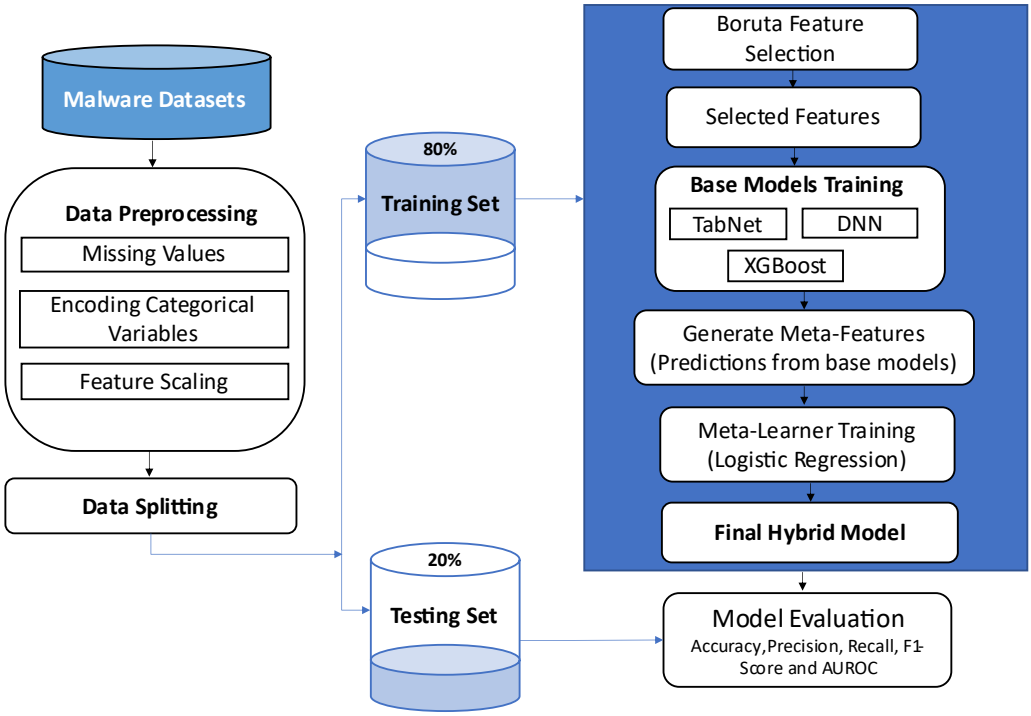

Figure 1. Proposed framework

Table 1. Statistical overview of the TUANDROMD dataset

Statistic	Value
Dataset shape	4,465 samples × 242 features
Malware instances	3,565
Goodware instances	899

3.1.2. CIC-AndMal2017 Dataset

The CIC-AndMal2017 dataset is a widely used benchmark for Android malware analysis, provided in CSV format for ease of use. It contains pre-extracted and curated features that facilitate the classification of multiple malware families as well as benign applications.

The dataset was derived from a publicly available Android malware repository, where relevant features were systematically extracted and organized to improve usability, reproducibility, and analytical consistency [33]. It comprises 426,735 samples described by 81 features. Among these, 150,382 instances correspond to malware (class 1), while 276,353 instances correspond to benign applications (class 0). A statistical summary of the dataset is provided in Table 2.

Table 2. Statistical overview of the CIC-AndMal2017 dataset.

Statistic	Value
Dataset shape	426,735 samples × 81 features
Malware instances	150,382
Benign instances	276,353

3.2. Data Preprocessing

Data preprocessing is performed to ensure consistency, reliability, and fairness in model training. The preprocessing pipeline includes missing value inspection, target encoding, feature scaling, and dataset splitting. Both datasets are already feature-engineered and do not

require additional feature extraction; instead, feature refinement is performed through Boruta-based selection in the subsequent stage.

3.2.1. Missing Values

The datasets were examined for missing values, and no systematic missing data patterns were identified. Therefore, no imputation techniques, such as mean or median substitution, were required.

3.2.2. Encoding Target Variables

The target variable preprocessing differs slightly between the two datasets. In the TUANDROMD dataset, labels were originally provided in textual form (malware and goodware). These labels were converted into numerical format using label encoding, where malware is encoded as 1 and goodware as 0, ensuring compatibility with ML algorithms. In contrast, the CIC-AndMal2017 dataset already provides the target variable in binary numerical form (1 for malware and 0 for benign), and therefore no additional encoding was required.

3.2.3. Feature Scaling

Feature scaling is applied to improve model stability and convergence, particularly for DL models such as DNN. In this study, standardization is performed using the StandardScaler technique, which transforms each feature to have zero mean and unit variance, as defined in Equation (1).

$$X' = \frac{X - \mu}{\sigma} \tag{1}$$

where X denotes the original feature value, μ is the mean, σ is the standard deviation, and X' is the standardized value. This transformation ensures that all features contribute proportionally during model training.

However, feature scaling is not applied to the TabNet model. TabNet incorporates an internal feature transformation mechanism through attentive feature selection and normalization, enabling it to effectively process raw tabular data without prior scaling.

3.3. Train-Test Split and Cross-Validation

The dataset is partitioned into training and testing subsets using an 80:20 split, with stratification applied to preserve the original class distribution. This ensures that both subsets maintain representative proportions of malware and benign samples. To improve robustness and reliability, k-fold cross-validation is employed on the training data during model development. In each fold, the training data is further divided into training and validation subsets. All preprocessing steps, including feature selection, are performed strictly within the training folds to prevent data leakage and ensure unbiased performance evaluation.

It is important to note that this study does not apply explicit class balancing techniques such as oversampling, undersampling, or synthetic data generation. Instead, the framework relies on feature optimization and ensemble learning to maintain stable performance under naturally imbalanced data distributions.

3.4. Boruta Feature Selection

To improve model performance and eliminate irrelevant or noisy features, a wrapper-based feature selection technique was employed using the Boruta algorithm. Feature selection plays a critical role in Android malware detection due to the high dimensionality and redundancy commonly present in static and dynamic feature sets, which can adversely affect model generalization and increase computational complexity, particularly in hybrid and DL frameworks.

In this study, Boruta was implemented using the BorutaPy library, with a Random Forest classifier configured with a maximum depth of 5, balanced class weights, and parallel processing enabled. The incorporation of class weighting allows the Random Forest estimator within Boruta to assign greater importance to minority-class samples during feature importance computation. This enhances the reliability of feature evaluation under imbalanced class distributions. However, it is important to emphasize that this mechanism operates solely at the feature importance estimation stage and does not modify the underlying data

distribution. Therefore, it should not be interpreted as an explicit class balancing or resampling strategy.

Boruta was selected over conventional feature selection methods due to its all-relevant feature selection capability. Unlike filter-based approaches, such as correlation analysis or ANOVA, which evaluate features independently, Boruta leverages the Random Forest algorithm to capture nonlinear relationships and interactions among features. Furthermore, unlike dimensionality reduction techniques such as Principal Component Analysis, which transform features into lower-dimensional representations at the expense of interpretability, Boruta preserves the original feature space while identifying all features that contribute meaningfully to the target variable.

The Boruta algorithm operates by iteratively comparing the importance of each original feature with that of corresponding randomized shadow features. Feature importance is typically computed using the mean decrease in impurity from the Random Forest model, as expressed in Equation (2) [34].

$$I(f_j) = \frac{1}{T}\sum_{t=1}^{T} \Delta i_t(f_j) \tag{2}$$

where $I(f_j)$ represents the importance score of features f_j, T is the total number of trees in the Random Forest, and $\Delta i_t(f_j)$ denotes the impurity decrease contributed by feature f_j in tree t. A feature is considered relevant if its importance score consistently exceeds the maximum importance achieved by the shadow features across iterations.

To ensure methodological rigor, the Boruta algorithm was trained exclusively on the training dataset, thereby preventing data leakage and ensuring unbiased feature selection. Following the fitting process, a subset of relevant features was identified based on the Boruta support mask. These selected features were subsequently used to transform both the training and testing datasets, resulting in reduced feature representations that retain only the most informative attributes.

The reduced feature sets were then used as input for both the individual base models (TabNet, XGBoost, and DNN) and the proposed hybrid stacking framework. By eliminating irrelevant and redundant features, this step contributes to improved model generalization, reduced computational complexity, and enhanced robustness across both individual and ensemble learning approaches.

3.5. Model Development

This study develops a hybrid stacking framework for Android malware detection by integrating multiple ML and DL models. The proposed architecture leverages the complementary strengths of three base learners TabNet, XGBoost, and a DNN whose outputs are combined using a Logistic Regression meta-learner.

3.5.1. Proposed Model

The proposed model adopts a two-level stacking ensemble architecture to enhance predictive performance through the integration of heterogeneous learning algorithms. At the first level, TabNet, XGBoost, and the DNN are trained independently using the Boruta-selected features. Each model generates probabilistic predictions for both training and testing samples.

TabNet is incorporated due to its attention-based feature selection mechanism, which enables sequential and adaptive learning of feature importance in tabular data. This capability allows it to capture feature interactions that may not be effectively modeled by tree-based methods or conventional neural networks. XGBoost contributes strong predictive performance through gradient boosting and robust handling of nonlinear relationships, while the DNN captures hierarchical and nonlinear feature representations. The combination of these models promotes diversity in learning behavior, which is essential for effective stacking. Notably, TabNet is included not for its standalone predictive superiority, but for its ability to introduce complementary decision boundaries and less correlated prediction errors, thereby enhancing ensemble diversity and robustness.

The TabNet model was trained for a maximum of 50 epochs with early stopping (patience = 10) to mitigate overfitting. The XGBoost model was configured with 300 estimators, a maximum depth of 6, and a learning rate of 0.05, with subsampling applied to improve generalization. The DNN architecture consists of two hidden layers with 128 and 64 neurons,

respectively, using ReLU activation functions and a sigmoid output layer for binary classification. Input features were standardized prior to DNN training to ensure stable convergence. The probabilistic outputs generated by the base learners are concatenated to form a stacked feature matrix:

$$S = [P_{TabNet}, P_{XGB}, P_{DNN}] \quad (3)$$

This stacked representation serves as input to a Logistic Regression meta-learner, which learns the optimal combination of base model predictions. The final prediction of the hybrid model is defined as:

$$\hat{y} = f_{LR}(P_{TabNet}, P_{XGB}, P_{DNN}) \quad (3)$$

In stacking ensembles, predictive performance is influenced not only by the strength of individual models but also by the diversity of their learned representations. Models with different inductive biases tend to produce complementary error patterns, enabling the meta-learner to achieve improved generalization. In this framework, TabNet contributes to ensemble diversity by providing complementary decision patterns, thereby supporting the robustness of the overall system. A summary of the stacking configuration and meta-learner setup is presented in Table 3.

Table 3. Stacking ensemble configuration and meta-learner setup

Component Level	Model/Operation	Configuration / Description
Level-1 Base Learner 1	TabNet	Trained on Boruta-selected features; outputs class probabilities
Level-1 Base Learner 2	XGBoost	Trained on top-ranked Boruta features; outputs class probabilities
Level-1 Base Learner 3	DNN	Trained on standardized feature set; outputs class probabilities
Feature Fusion	Horizontal concatenation	Combines base predictions into stacked feature matrix
Level-2 Meta-Learner	Logistic Regression	Trained on stacked features to produce final predictions

3.5.2. Baseline Models

To evaluate the effectiveness of the proposed hybrid framework, each base learner—TabNet, XGBoost, and the DNN was also trained as a standalone baseline model. All baseline models were trained using the same Boruta-selected features and identical data partitioning strategy to ensure a fair comparison. This consistent setup ensures that performance differences can be attributed to the modeling approach rather than variations in preprocessing or training conditions. The baseline models were evaluated on the same test set using Accuracy, Precision, Recall, F1-score, and AUROC. A summary of the baseline configurations is provided in Table 4.

Table 4. Baseline model configuration summary.

Model	Key Parameters
TabNet	max_epochs = 50, patience = 10, batch_size = 1024, virtual_batch_size = 128
XGBoost	n_estimators = 300, max_depth = 6, learning_rate = 0.05, subsample = 0.8, colsample_bytree = 0.8
DNN	Hidden layers: 128, 64; activation = ReLU; optimizer = Adam; epochs = 20; batch_size = 256

3.6. Model Evaluation

To ensure a fair and consistent evaluation, both the hybrid model and baseline models were trained and tested using the same Boruta-selected feature set and data partitioning scheme. The dataset was divided into training and testing subsets using an 80:20 stratified split to preserve class distribution. Model performance was evaluated using standard

classification metrics, including Accuracy, Precision, Recall, F1-score, and Area Under the Receiver Operating Characteristic Curve (AUROC). Particular emphasis is placed on recall and F1-score, as these metrics are critical in malware detection scenarios where minimizing false negatives is essential.

4. Results and Discussion

This section presents the experimental results and analysis of the proposed hybrid stacking model for Android malware detection using the TUANDROMD and CIC-AndMal2017 datasets. Model performance is evaluated using standard classification metrics, including Accuracy, Precision, Recall, F1-score, AUROC, and confusion matrices. The evaluation focuses on assessing the robustness and generalization capability of the proposed hybrid framework in comparison with the baseline models (TabNet, XGBoost, and DNN).

4.1. Feature Selection

Feature selection was performed on both datasets using the Boruta algorithm to identify the most informative attributes for malware detection. This process aims to reduce dimensionality, eliminate irrelevant features, and improve model efficiency and generalization.

4.1.1. TUANDROMD Dataset

For the TUANDROMD dataset, Boruta was executed for 100 iterations over the original 241 features. In the initial iterations, all features were labeled as tentative, indicating uncertainty in their importance. By iteration 8, 98 features were confirmed as relevant, 14 remained tentative, and 129 were rejected. As the process continued, the number of confirmed features increased while tentative features decreased. At the final iteration, 104 features were confirmed, 1 remained tentative, and 136 were rejected. These results indicate that approximately half of the original features contribute meaningfully to the classification task, enabling the model to focus on the most discriminative attributes.

Table 5. Boruta feature selection summary for TUANDROMD dataset

Iteration	Confirmed Features	Tentative Features	Rejected Features
1–7	0	241	0
8–99	98–104	14–3	129–134
100	104	1	136

4.1.2. CIC-AndMal2017 Dataset

A similar iterative process was observed for the CIC-AndMal2017 dataset. Initially, all 80 features were classified as tentative. By iteration 8, 68 features were confirmed and 12 were rejected, with no remaining tentative features in subsequent iterations. At the end of the process, 68 features were retained as relevant for classification. This result suggests that a reduced subset of features captures the essential information required for accurate malware detection, thereby reducing noise and mitigating potential overfitting.

Table 6. Boruta feature selection summary for CIC-AndMal2017 dataset

Iteration	Confirmed Features	Tentative Features	Rejected Features
1–7	0	80	0
8–100	68	0	12

Overall, Boruta-based feature selection reduces dimensionality in both datasets, retaining 104 features for TUANDROMD and 68 features for CIC-AndMal2017. These selected features are used as input for both the baseline models and the proposed hybrid stacking framework, ensuring that subsequent evaluations are performed on the most informative feature sets.

4.2. Experimental Results on TUANDROMD Dataset

The performance of the proposed hybrid stacking model and baseline models on the TUANDROMD dataset is presented in Table 7. All models were evaluated using consistent

metrics to ensure a fair comparison. The results indicate that the proposed hybrid stacking model achieves the strongest overall performance. While XGBoost and DNN exhibit highly competitive results, the hybrid model provides a slightly improved and more consistent balance between precision and recall, suggesting enhanced generalization through ensemble integration.

Table 7. Performance comparison on TUANDROMD dataset

Model	Accuracy	Precision	Recall	F1 Score	AUROC
TabNet	0.855543	0.852657	0.990182	0.916288	0.974170
XGBoost	0.996641	0.995810	1.000000	0.997901	0.999899
DNN	0.994401	0.995798	0.997195	0.996496	0.999782
Proposed Hybrid Stacking	0.997760	0.997203	1.000000	0.998599	0.999423

The confusion matrix in Figure 2 shows that the hybrid model correctly identifies all malware instances (zero false negatives) and misclassifies only two benign samples as malware. This behavior indicates strong detection capability while maintaining a low false-positive rate. The baseline models exhibit similar trends but with minor differences. XGBoost and DNN achieve near-perfect classification with only a small number of misclassifications. In contrast, TabNet demonstrates a noticeably higher false-positive rate, indicating weaker discrimination of benign samples despite maintaining high recall.

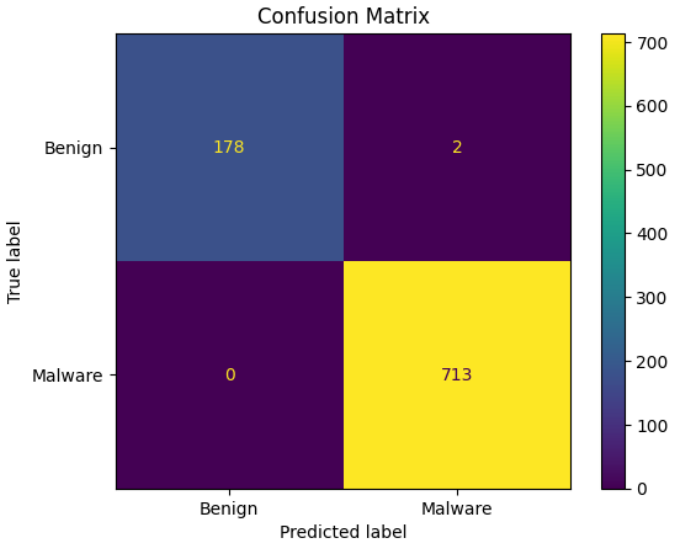

Figure 2. Confusion matrix of the hybrid model on TUANDROMD dataset

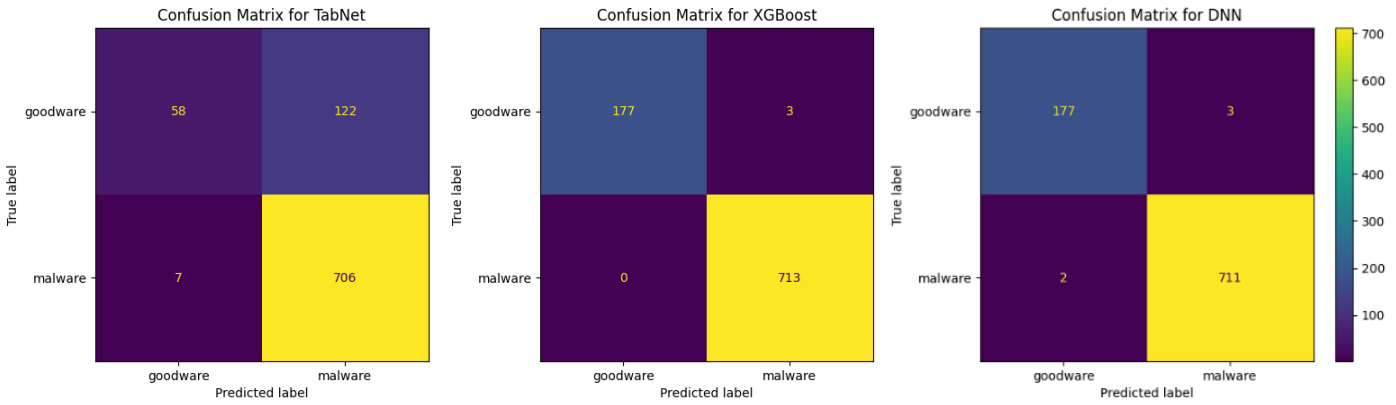

Figure 3. Confusion matrices of baseline models on TUANDROMD dataset

Figure 4 summarizes the performance across all evaluation metrics and confirms the trends observed in Table 7. The hybrid model maintains consistently strong performance across all metrics, indicating stable behavior across different evaluation criteria.

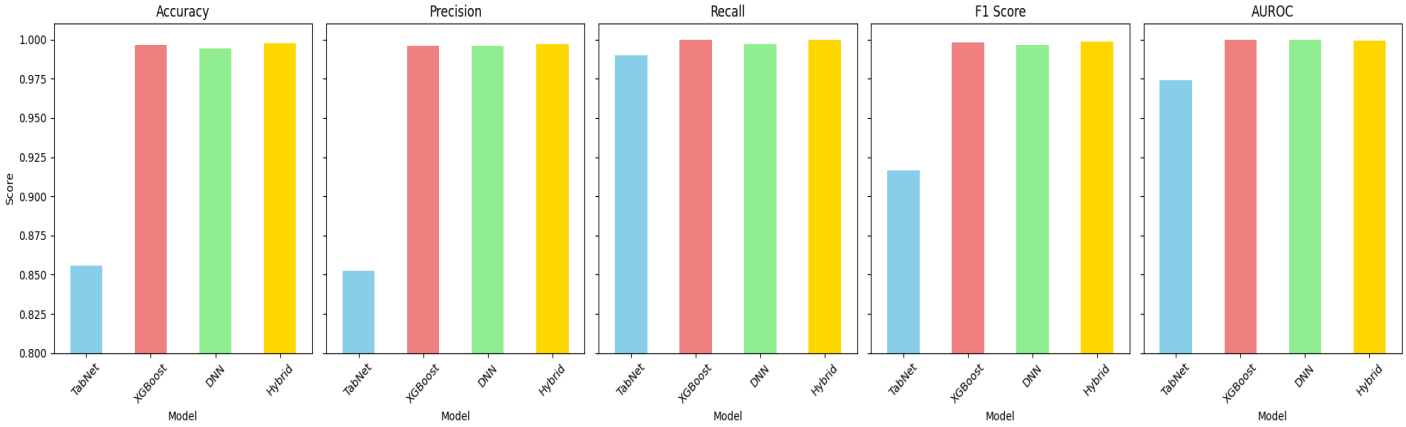

Figure 4. Comparative performance of all models on TUANDROMD dataset

4.3. Experimental Results on CIC-AndMal2017 Dataset

The performance of the proposed hybrid stacking model and the baseline models (TabNet, XGBoost, and DNN) on the CIC-AndMal2017 dataset is presented in Table 8. All models were evaluated using consistent performance metrics, including Accuracy, Precision, Recall, F1-score, and AUROC, to ensure a fair and comprehensive comparison.

Table 8. Performance comparison on CIC-AndMal2017 dataset

Model	Accuracy	Precision	Recall	F1 Score	AUROC
TabNet	0.855543	0.852657	0.990182	0.916288	0.974170
XGBoost	0.996641	0.995810	1.000000	0.997901	0.999899
DNN	0.994401	0.995798	0.997195	0.996496	0.999782
Proposed Hybrid Stacking	0.978945	0.969766	0.970508	0.970137	0.996030

The results indicate that the proposed hybrid stacking model achieves competitive and well-balanced performance across all evaluation metrics. While XGBoost attains the highest individual metric values, the hybrid framework demonstrates a more consistent balance between precision and recall, indicating stable classification behavior under this dataset.

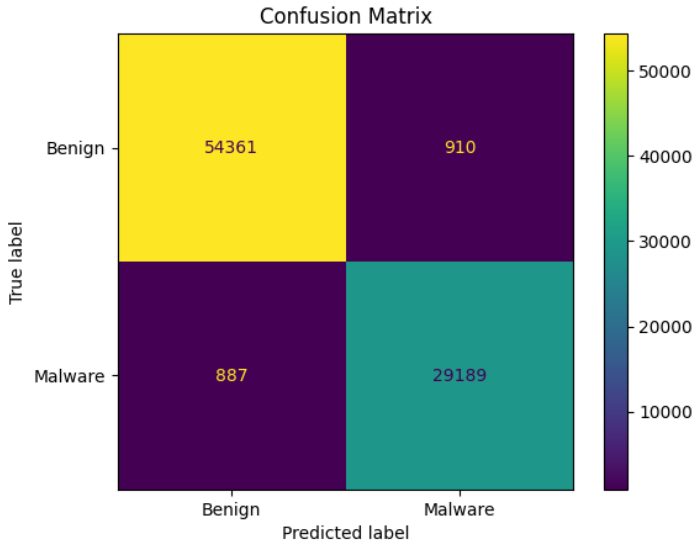

Figure 5. Confusion matrix of the hybrid model on CIC-AndMal2017 dataset

The confusion matrix in Figure 5 provides further insight into the classification performance of the proposed model. The hybrid framework correctly classifies 54,361 benign samples and 29,189 malware samples, with 910 false positives and 887 false negatives. This distribution reflects a balanced error profile, indicating that the model maintains both detection capability and false alarm control.

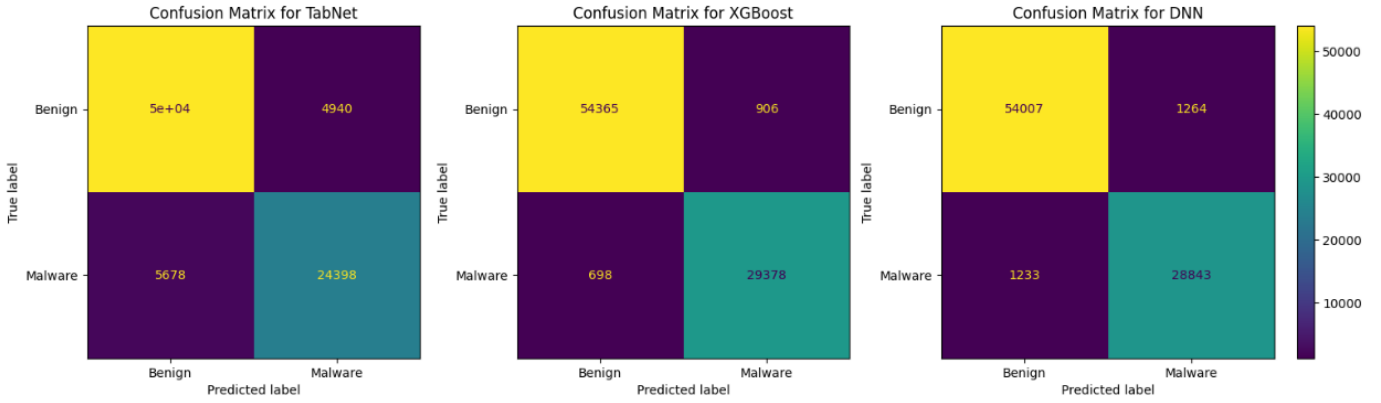

Figure 6. Confusion matrices of baseline models on CIC-AndMal2017 dataset

In comparison, the baseline models exhibit more varied performance characteristics. XGBoost demonstrates strong classification capability with minimal misclassifications, while the DNN also performs competitively with slightly higher error rates. In contrast, TabNet produces a substantially higher number of misclassifications, particularly in both false positives and false negatives, indicating lower effectiveness on this dataset. These observations are consistent with the quantitative results presented in Table 8.

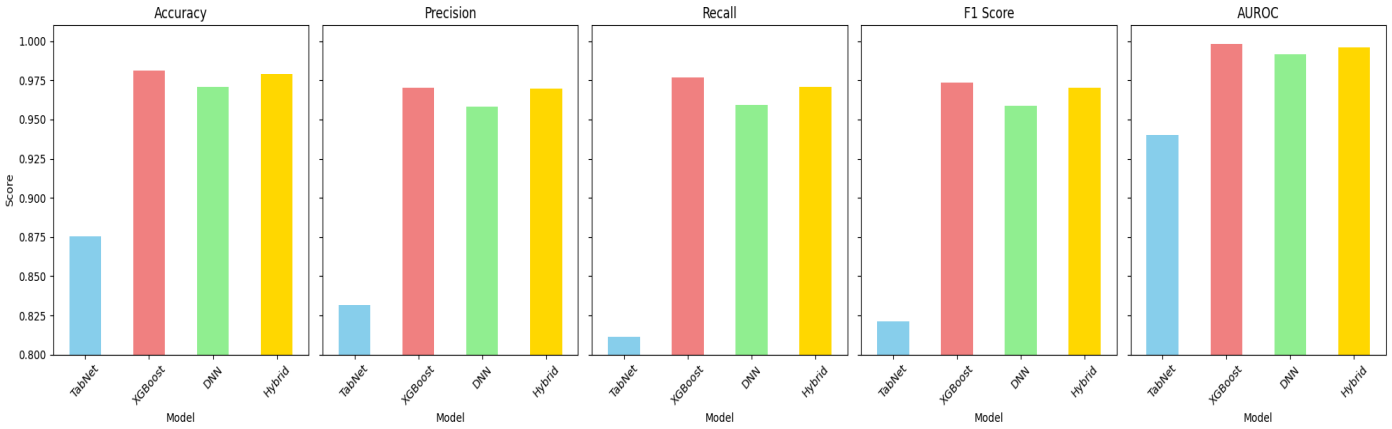

Figure 7. Comparative performance of all models on CIC-AndMal2017 dataset

Figure 7 provides a visual summary of the evaluation metrics and confirms the performance trends observed in Table 8. The hybrid model maintains consistent performance across all metrics, reflecting stable behavior relative to the baseline models.

4.4. Extended Evaluation

4.4.1. Ablation Study

To assess the contribution of Boruta-based feature selection to the overall performance of the proposed framework, an ablation study was conducted by training all models using the original feature set without applying feature selection. The same preprocessing pipeline, model configurations, and train–test split were maintained to ensure a fair comparison. The results indicate that while the proposed hybrid stacking model maintains strong performance without feature selection, its effectiveness is consistently lower than the Boruta-enhanced version reported in Sections 4.2 and 4.3. A slight reduction in accuracy, precision, and F1-

score is observed across both datasets, suggesting that redundant and less informative features negatively affect model generalization.

Table 9. Model performance without Boruta feature selection on TUANDROMD dataset

Model	Accuracy	Precision	Recall	F1 Score	AUROC
TabNet	0.8914	0.8802	1.0000	0.9363	0.9740
XGBoost	0.9866	0.9848	0.9986	0.9916	0.9977
DNN	0.9910	0.9958	0.9930	0.9944	0.9996
Proposed Hybrid	0.9955	0.9944	1.0000	0.9972	0.9998

Table 10. Model performance without Boruta feature selection on CIC-AndMal2017 dataset

Model	Accuracy	Precision	Recall	F1 Score	AUROC
TabNet	0.9346	0.8826	0.9398	0.9103	0.9606
XGBoost	0.9755	0.9548	0.9769	0.9657	0.9965
DNN	0.9428	0.9249	0.9120	0.9184	0.9683
Proposed Hybrid	0.9755	0.9589	0.9722	0.9655	0.9966

The impact is more pronounced in individual models, particularly TabNet and the DNN, which exhibit increased variability and reduced precision when trained on the full feature set. This indicates higher sensitivity to noise and irrelevant features. In contrast, Boruta-based feature selection improves feature quality by removing redundancy, enabling all base learners to operate on a more discriminative and stable feature space.

4.4.2. Comparison with State-of-the-Art Approaches

Benchmarking against recent state-of-the-art methods further contextualizes the performance of the proposed framework. On the TUANDROMD dataset, prior studies reported accuracies ranging from 97% to 99.3%, with F1-scores and AUROC values generally below 0.9998. For instance, Masari et al. [13] achieved an accuracy of 99.30%, F1-score of 99.60%, and AUROC of 0.9998, which are slightly lower than the results obtained by the proposed hybrid model (accuracy: 99.78%, F1-score: 99.86%, AUROC: 0.9994).

Table 11. State-of-the-art comparison on TUANDROMD dataset

Reference	Accuracy	Precision	Recall	F1-score	ROC AUC
Wahajat et al. [35]	0.9860	0.9890	0.9940	–	–
Wahajat et al. [36]	0.9890	0.9700	1.0000	0.8420	–
Kacem and Tossou [26]	0.9925	0.9926	0.9925	0.9926	0.9876
Palabaş [37]	0.9700	0.8500	–	–	–
Etem [17]	–	–	–	–	–
Masari et al. [13]	0.9930	0.9920	1.0000	0.9960	0.9998
Proposed Hybrid	0.9978	0.9972	1.0000	0.9986	0.9994

On the CIC-AndMal2017 dataset, prior studies such as Museeb et al. [36], [37] and Rekik et al. [35] reported accuracies ranging from 91.2% to 96.2%, generally accompanied by lower F1-scores and less balanced precision–recall trade-offs. In comparison, the proposed hybrid model achieves an accuracy of 97.89% and an F1-score of 97.01%, indicating competitive and stable performance across evaluation metrics.

Table 12. State-of-the-art comparison on CIC-AndMal2017 dataset

Reference	Accuracy	Precision	Recall	F1-score	ROC AUC
Museeb et al. [38]	0.9500	0.9600	0.9500	0.9500	–
Museeb et al. [39]	0.9120	0.9113	0.8409	0.9190	–
Rekik et al. [40]	0.9620	0.9580	0.9660	0.9620	–
Proposed Hybrid	0.9789	0.9698	0.9705	0.9701	0.9960

These results suggest that the proposed hybrid stacking framework achieves competitive performance relative to existing approaches, particularly in terms of maintaining a balanced precision–recall trade-off and stable generalization across datasets.

4.5. Discussion and Findings

4.5.1. Mechanism of Performance Improvement

The consistent performance observed across both datasets can be attributed to the interaction between Boruta-based feature selection and heterogeneous ensemble learning. Boruta reduces feature redundancy and noise, ensuring that all base models operate on a compact and informative feature space, thereby improving generalization and reducing overfitting. In addition, the ablation results indicate that this feature optimization step provides a relatively consistent contribution across models, particularly for learning approaches that are more sensitive to irrelevant or redundant features.

Within the ensemble, each base learner contributes a distinct inductive bias. XGBoost effectively models structured nonlinear relationships, while the DNN captures deeper hierarchical feature interactions. TabNet, in contrast, introduces attention-based feature selection at the instance level, enabling adaptive emphasis on relevant features, thereby increasing model diversity. Although TabNet exhibits lower standalone performance compared to XGBoost and the DNN, its role within the ensemble is not primarily based on individual predictive strength. Rather, TabNet contributes to diversity by producing complementary prediction patterns that differ from those of tree-based and conventional DL models. This diversity enables the meta-learner to reduce correlated errors and improve overall robustness.

4.5.2. Performance Interpretation and Trade-offs

The experimental results indicate that the hybrid framework consistently achieves a balanced trade-off between precision and recall across both datasets. While XGBoost demonstrates the strongest standalone performance, the hybrid model provides more stable behavior in terms of F1-score and AUROC, indicating improved generalization. It is also observed that the performance improvement of the hybrid model over XGBoost is relatively modest in several metrics. This suggests that when strong individual learners are already capable of capturing the underlying data structure, the contribution of stacking may be more evident in stabilizing predictions and balancing error distribution rather than significantly increasing overall accuracy.

In malware detection, recall is critical for minimizing false negatives, as undetected malware poses significant security risks. At the same time, precision is essential to control false alarms and maintain system usability. The hybrid model maintains high recall while preserving precision, resulting in a balanced F1-score that reflects reliable detection performance. Although the numerical performance gains appear small, even slight reductions in false positives or false negatives can have meaningful implications in practical security systems. While the stacking framework introduces additional computational complexity, this overhead remains manageable in offline training environments and is justified by improved robustness and reduced model bias.

4.5.3. Robustness under Imbalanced Data and Limitations

Importantly, the proposed framework achieves stable performance without employing explicit resampling or synthetic class balancing techniques. Instead, the observed improvements are attributed to feature optimization and class-aware feature evaluation through weighting during the Boruta stage, combined with the complementary learning behavior of the heterogeneous stacking ensemble. This distinction is important because the framework focuses on improving generalization and detection stability under imbalanced data conditions rather than altering the original class distribution.

At the same time, the results suggest that dataset characteristics play an important role in determining the extent of performance improvement. The TUANDROMD and CIC-AndMal2017 datasets exhibit relatively well-structured and partially separable feature spaces due to prior feature engineering. Under such conditions, strong models such as XGBoost are already able to achieve high performance, which may limit the additional gains obtained from more complex ensemble strategies.

Despite the strong performance achieved by the proposed model across both datasets, it is important to critically consider the possibility of data separability and overfitting. The

high classification scores may partially reflect the inherent separability of the datasets, where malware and benign samples exhibit distinguishable feature patterns. In such cases, ML models may achieve near-perfect performance without necessarily reflecting real-world complexity.

To mitigate overfitting concerns, several safeguards were applied in this study, including the use of Boruta-based feature selection, stratified train-test splitting, and evaluation on two independent benchmark datasets. Additionally, the inclusion of heterogeneous base learners introduces model diversity, which reduces the likelihood of memorizing dataset-specific patterns. However, further validation on more diverse and real-world datasets is necessary to fully confirm the generalization capability of the proposed approach.

5. Conclusions

This study presents an optimized hybrid stacking framework for Android malware detection that integrates TabNet, XGBoost, and a Deep Neural Network as heterogeneous base learners, combined through a Logistic Regression meta-learner and supported by Boruta-based feature selection. Experimental evaluation on the TUANDROMD and CIC-AndMal2017 datasets demonstrates that the proposed framework achieves consistently strong performance, with high accuracy and reliable precision–recall balance across both datasets. The findings indicate that the effectiveness of the proposed approach is not solely driven by model complexity but is largely influenced by feature quality. In particular, Boruta-based feature selection provides a consistent contribution by reducing redundancy and improving the discriminative capacity of the input space. In contrast, the stacking mechanism contributes primarily by enhancing prediction stability and balancing error distribution rather than delivering substantial gains over strong individual models such as XGBoost. This observation highlights that the synergy between feature optimization and heterogeneous learning plays a more critical role than any single model component.

These results support the research hypothesis that combining feature optimization with a heterogeneous stacking framework improves detection performance and stability in Android malware classification. More importantly, the study demonstrates that robust performance can be achieved without relying on explicit class balancing or synthetic resampling techniques, instead leveraging feature-level optimization and complementary learning behavior to handle imbalanced data conditions effectively. From a broader perspective, the proposed framework contributes to the field by providing a practical and scalable approach for malware detection that emphasizes robustness, balanced classification behavior, and generalization across datasets. The findings also offer insight into the relative importance of feature selection and ensemble design, suggesting that improving feature quality may be more impactful than increasing model complexity in structured malware detection tasks.

However, several limitations should be acknowledged. The evaluation is limited to two benchmark datasets, which may not fully represent the variability and evolving characteristics of real-world Android malware. Additionally, the stacking framework introduces increased computational complexity and may require careful hyperparameter tuning, particularly for components such as TabNet. The study also does not explicitly address real-time deployment constraints, which are critical in operational security environments. Future work will focus on validating the proposed framework on more diverse and large-scale real-world datasets, exploring more computationally efficient model variants, and integrating explainable AI techniques to improve interpretability. Extending the framework to real-time and federated learning settings also represents a promising direction for enhancing its applicability in distributed and privacy-sensitive malware detection systems.

Author Contributions: Conceptualization: M.A.D., M.S.M., U.P.C., and A.A.; Methodology: A.A.; Software: A.A.; Validation: M.A.D., M.S.M., U.P.C., and A.A.; Formal analysis: A.A.; Investigation: M.A.D., M.S.M., U.P.C., and A.A.; Resources: M.A.D., M.S.M., U.P.C., and A.A.; Data curation: A.A.; Writing—original draft preparation: A.A.; Writing—review and editing: A.A.; Visualization: A.A.; Supervision: M.A.D., M.S.M., U.P.C., and A.A.; Project administration: A.A.; Funding acquisition: M.A.D., M.S.M., U.P.C., and A.A.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Data Availability Statement: The dataset is publicly available online and can be accessed via the link; https://archive.ics.uci.edu/dataset/855/tuandromd+%28tezpur+university+android+malware+dataset%29 and https://data.mendeley.com/datasets/bpkksc9v5s/5.

Acknowledgments: The authors would like to acknowledge Abdulrashid Abdulrauf for his significant contribution to the experimental analysis and model development stages of this study.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] D. Elagina, "Number of smartphone users worldwide from 2016 to 2029," *Statista*, 2026. https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/

[2] Statista Research Department, "Global market share held by mobile operating systems from 2009 to 2025," *Statista*, 2026. https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/

[3] M. Benmalek, "Ransomware on cyber-physical systems: Taxonomies, case studies, security gaps, and open challenges," *Internet Things Cyber-Physical Syst.*, vol. 4, pp. 186–202, 2024, doi: 10.1016/j.iotcps.2023.12.001.

[4] M. Cen, F. Jiang, X. Qin, Q. Jiang, and R. Doss, "Ransomware early detection: A survey," *Comput. Networks*, vol. 239, p. 110138, Feb. 2024, doi: 10.1016/j.comnet.2023.110138.

[5] M. Baykara and E. Colak, "A review of cloned mobile malware applications for android devices," in *2018 6th International Symposium on Digital Forensic and Security (ISDFS)*, Mar. 2018, pp. 1–5. doi: 10.1109/ISDFS.2018.8355388.

[6] M. Conti, T. Dargahi, and A. Dehghantanha, "Cyber Threat Intelligence: Challenges and Opportunities," in *Advances in Information Security*, 2018, pp. 1–6. doi: 10.1007/978-3-319-73951-9_1.

[7] A. Pagan and K. Elleithy, "A Multi-Layered Defense Approach to Safeguard Against Ransomware," in *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, Jan. 2021, pp. 0942–0947. doi: 10.1109/CCWC51732.2021.9375988.

[8] B. Yamany, M. S. Elsayed, A. D. Jurcut, N. Abdelbaki, and M. A. Azer, "A Holistic Approach to Ransomware Classification: Leveraging Static and Dynamic Analysis with Visualization," *Information*, vol. 15, no. 1, p. 46, Jan. 2024, doi: 10.3390/info15010046.

[9] M. M. Raza *et al.*, "Malware Detection and AI Integration: A Systematic Review of Current Trends and Future Directions," *Comput. Model. Eng. Sci.*, vol. 146, no. 3, pp. 1–10, 2026, doi: 10.32604/cmes.2025.074164.

[10] V. Kouliaridis and G. Kambourakis, "A Comprehensive Survey on Machine Learning Techniques for Android Malware Detection," *Information*, vol. 12, no. 5, p. 185, Apr. 2021, doi: 10.3390/info12050185.

[11] D. Gibert, C. Mateu, and J. Planes, "The rise of machine learning for detection and classification of malware: Research developments, trends and challenges," *J. Netw. Comput. Appl.*, vol. 153, p. 102526, Mar. 2020, doi: 10.1016/j.jnca.2019.102526.

[12] M. A. Rahman, G. A. Francia, and H. Shahriar, "Leveraging GANs for Synthetic Data Generation to Improve Intrusion Detection Systems," *J. Futur. Artif. Intell. Technol.*, vol. 1, no. 4, pp. 429–439, Feb. 2025, doi: 10.62411/faith.3048-3719-52.

[13] M. S. Masari, M. A. Danladi, I. L. Onyinye, and L. K. Tohomdet, "Android Malware Detection Using Machine Learning with SMOTE-Tomek Data Balancing," *J. Comput. Theor. Appl.*, vol. 3, no. 3, pp. 302–313, Jan. 2026, doi: 10.62411/jcta.15084.

[14] A. K. Jilani and S. Samreen, "Ensemble Based Android Malware Detection Using a Bio-Inspired Meta-Heuristic Feature Engineering Approach," in *2024 International Conference on Computing, Internet of Things and Microwave Systems (ICCIMS)*, Jul. 2024, pp. 1–6. doi: 10.1109/ICCIMS61672.2024.10690649.

[15] M. Ibadullah, S. A. Amalina, W. Ghozi, and F. A. Rafrastara, "Machine Learning-based Malware Detection on Android Operating System using AdaBoost Algorithm and ReliefF Feature Selection Method," in *2024 International Seminar on Application for Technology of Information and Communication (iSemantic)*, Sep. 2024, pp. 359–364. doi: 10.1109/iSemantic63362.2024.10762096.

[16] Y. Pristyanto, A. F. Nugraha, B. Wulansari, M. Sulistiyono, and A. Sunyoto, "Enhancement of Machine Learning Models Using Intersection Filtering Model Based on Recursive Feature Elimination on Specified Android Malware Classification," *Int. J. Intell. Eng. Syst.*, vol. 18, no. 3, pp. 836–852, Apr. 2025, doi: 10.22266/ijies2025.0430.56.

[17] T. Etem, "Comparative Analysis of Principle Component Analysis and Anova Feature Selection in Malware Detection," *Firat Univ. J. Exp. Comput. Eng.*, vol. 5, no. 1, pp. 299–315, Feb. 2026, doi: 10.62520/fujece.1635121.

[18] A. Fenjan, D. J. Md Desa, D. S. Elaskari, A. Theeb, and A. S. Alsayafi, "Addressing Data Imbalance in Malware Detection: A Comparison Analysis of Machine Learning Models for Malware Detection on Balanced and Imbalanced Datasets," in *Proceedings of the Sixth International Conference on Digital Age & Technological Advances for Sustainable Development*, May 2025, pp. 1–5. doi: 10.1145/3747897.3747898.

[19] S. Al Ali *et al.*, "Android Malware Detection Using Machine Learning," in *2024 IEEE International Conference on Internet of Things and Intelligence Systems (IoTaIS)*, Nov. 2024, pp. 79–84. doi: 10.1109/IoTaIS64014.2024.10799339.

[20] E. S. Akkaya and E. V. Altay, "Investigating the Performance of Machine Learning Methods for Malware Detection," in *EAI/Springer Innovations in Communication and Computing*, 2025, pp. 329–340. doi: 10.1007/978-3-031-88999-8_25.

[21] H. Abdulla, "Android Malware Detection: A Machine Leaning Approach," *arXiv*. Nov. 02, 2025. [Online]. Available: http://arxiv.org/abs/2511.00894

[22] M. K. Shah, "AI-Based Framework for Ransomware Detection in Android Systems: Enhancing Mobile Security," in *2025 5th International Conference on Artificial Intelligence and Signal Processing (AISP)*, Nov. 2025, pp. 1–8. doi: 10.1109/AISP68263.2025.11396254.

[23] N. G. Ambekar, N. N. Devi, S. Thokchom, and Yogita, "TabLSTMNet: enhancing android malware classification through integrated attention and explainable AI," *Microsyst. Technol.*, vol. 31, no. 3, pp. 695–713, Mar. 2025, doi: 10.1007/s00542-024-05615-0.

[24] Haris Mehmood, Muhammad Kamran Abid, Muhammad Fuzail, Ahmad Naeem, and Naeem Aslam, "Enhancing Real-time Android Malware Detection using Deep Learning and Fuzzy Logic-based Hybrid Models," *Kashf J. Multidiscip. Res.*, vol. 2, no. 06, pp. 161–170, Jun. 2025, doi: 10.71146/kjmr494.

[25] A. K. J, N. Das Patel, S. D, and A. Patel, "Behavioral Malware Detection via API Call Sequences: A Comparative Study of LSTM and Transformer Architectures Using NLP-Inspired Representations," *J. Comput. Theor. Appl.*, vol. 3, no. 4, pp. 443–456, Apr. 2026, doi: 10.62411/jcta.15811.

[26] T. Kacem and S. Tossou, "Trandroid: An Android Mobile Threat Detection System Using Transformer Neural Networks," *Electronics*, vol. 14, no. 6, p. 1230, Mar. 2025, doi: 10.3390/electronics14061230.

[27] M. Arya, S. Arya, and S. Arya, "H2O AutoML for Malware Detection: A Signal Processing Perspective on Accuracy, Efficiency, and Interpretability," in *2025 IEEE International Conference on Signals and Systems (ICSigSys)*, Nov. 2025, pp. 77–83. doi: 10.1109/ICSigSys67277.2025.11269166.

[28] M. R. Abdellah *et al.*, "Enhanced Federated Learning Framework Based on Deep Learning and Neutrosophic Set for Android Malware Classification," *Neutrosophic Sets Syst.*, vol. 82, no. 1, 2025, [Online]. Available: https://digitalrepository.unm.edu/cgi/viewcontent.cgi?article=3189&context=nss_journal

[29] A. Daulay, K. Ramli, R. Harwahyu, T. Hidayat, and B. Pranggono, "Novel Federated Graph Contrastive Learning for IoMT Security: Protecting Data Poisoning and Inference Attacks," *Mathematics*, vol. 13, no. 15, p. 2471, Jul. 2025, doi: 10.3390/math13152471.

[30] D. Mughole Kalimumbalo *et al.*, "SecFedMDM-1: A Federated Learning-Based Malware Detection Model for Interconnected Cloud Infrastructures," *IEEE Access*, vol. 13, pp. 101246–101261, 2025, doi: 10.1109/ACCESS.2025.3577706.

[31] R. B. Coulibaly, T. F. Bissyandé, R. Kafando, A. Sabané, and A. K. Kabore, "Privacy-Preserving Android Malware Detection Using Deep Federated Learning," in *2025 Cybersecurity4D (C4D)*, Aug. 2025, pp. 1–7. doi: 10.1109/C4D65382.2025.11306458.

[32] P. Borah, D. Bhattacharyya, and J. Kalita, "Malware Dataset Generation and Evaluation," in *2020 IEEE 4th Conference on Information & Communication Technology (CICT)*, Dec. 2020, pp. 1–6. doi: 10.1109/CICT51604.2020.9312053.

[33] M. S. Haque, M. S. Hossain, R. A. Robin, A. S. Tarisha, S. Ahmmed, and J. N. Mukta, "Android Malware Analysis." Mendeley Data, 2026. doi: 10.17632/bpkksc9v5s.5.

[34] M. B. Kursa and W. R. Rudnicki, "Feature Selection with the Boruta Package," *J. Stat. Softw.*, vol. 36, no. 11, 2010, doi: 10.18637/jss.v036.i11.

[35] A. Wajahat *et al.*, "An effective deep learning scheme for android malware detection leveraging performance metrics and computational resources," *Intell. Decis. Technol.*, vol. 18, no. 1, pp. 33–55, Feb. 2024, doi: 10.3233/IDT-230284.

[36] A. Wajahat *et al.*, "Outsmarting Android Malware with Cutting-Edge Feature Engineering and Machine Learning Techniques," *Comput. Mater. Contin.*, vol. 79, no. 1, pp. 651–673, 2024, doi: 10.32604/cmc.2024.047530.

[37] T. Palabaş, "Android malware classification using basic machine learning methods," *Adıyaman Üniversitesi Mühendislik Bilim. Derg.*, vol. 11, no. 23, pp. 190–202, Aug. 2024, doi: 10.54365/adyumbd.1462488.

[38] A. Museeb, Y. Hamed, Y. Baashar, A. M. Jamal Kanaan-Jebna, A. Amazigh Hamza, and R. Sokkalingam, "Android Malware Detection Using API Calls and Permissions With Random Forest Classifier," *IEEE Access*, vol. 14, pp. 6464–6480, 2026, doi: 10.1109/ACCESS.2026.3651861.

[39] A. Museeb, Y. Hamed, and H. Louis Tan, "Hybrid Random Forest and XGBoost Approach for Android Malware Detection," *Res. J. Maaref Univ. Appl. Sci.*, vol. 2, no. 1, p. 6, Feb. 2026, doi: 10.66422/wh0h4a90.

[40] S. Rekik, S. Mehmood, and M. Alkhonaini, "AdaptivePixGuard: Attention-Enhanced Temporal Convolutions for Robust Mobile Pixnapping Detection," *IEEE Access*, vol. 14, pp. 34072–34095, 2026, doi: 10.1109/ACCESS.2026.3669496.