e-ISSN: 3023-7343

TETHYS ENVIRONMENTAL SCIENCE
an International Journal

Identifying the Dominant Attribute in Android-Based Malware Detection

Neslihan Doğan*

Iskenderun Technical University, Department of Computer Engineering, 31220 Iskenderun, Hatay, Türkiye.

Research Article

Citation: Doğan, N. (2025). Identifying the Dominant Attribute in Android-Based Malware Detection. *Tethys Env. Sci*. 2(4): 199-205.

DOI: 10.5281/zenodo.18075972

Received: 09 September 2025

Accepted: 23 October 2025

Available Online: 28 December 2025

Publication Date: 29 December 2025

Copyright

2025 Doğan

Distributed Under

CC-BY 4.0

OPEN ACCESS

Abstract

This study presents a machine learning workflow developed for the detection and classification of Android-based ransomware using the TUANDROMD dataset. The workflow constitutes an integrated process that begins with the automated retrieval and extraction of data. During preprocessing, consistency of the target variable was ensured, and the dataset was partitioned into training (80%) and testing (20%) subsets while maintaining class balance. To prevent data leakage and enhance reproducibility, the model architecture was structured within a pipeline framework. This pipeline sequentially incorporates Feature Scaling, Feature Selection, and Classification stages. To improve the generalization capability of the model, hyperparameters such as the number of trees (*n_estimators*) and maximum depth (*max_depth*) were optimized using GridSearch with three-fold cross-validation. The best-performing model was subsequently evaluated on the reserved test set through a classification report and confusion matrix. Finally, the resulting model was serialized and stored in *.pkl* format for future use.

Keywords: *Malware, Android malware, feature selection, random forest.*

Introduction

As of today, the Android Operating System holds approximately 71-72% of the global mobile market share, making it a major target for cybercriminals. Traditional signature-based antivirus solutions have proven inadequate, as they can be easily bypassed through techniques such as obfuscation, polymorphism, and zero-day exploits. This, in turn, has prompted researchers to shift their focus towards more intelligent, adaptive, and low-false alarm dynamic solutions based on Machine Learning (ML) and Deep Learning (DL) approaches (Bensaoud et al., 2024; Smmarwar et al., 2024). Machine Learning methods employed in this domain analyze both static and dynamic

Correspondence: Neslihan Doğan
neslihandogan.lee24@iste.edu.tr

features extracted from Android applications, including permissions, API calls, system events, and network traffic. These approaches can detect previously unseen malware with high accuracy. However, in scenarios where the number of features is excessively large, models tend to become complex and computationally expensive, thereby hindering real-time detection on mobile devices. Current research therefore emphasizes the development of efficient models that maintain high accuracy while relying on a reduced number of features.

A review of the literature reveals several notable contributions. Borah et al. (2020) introduced the TUANDROMD dataset, a reliable resource frequently used in Android malware research, comprising 241 features and 4465 real-world samples. Wajahat et al. (2024) applied data cleaning and normalization techniques to the TUANDROMD dataset and achieved 99.0% accuracy using the Random Forest (RF) algorithm. Similarly, Polatidis et al. (2024) demonstrated that in their FSSDroid dataset, which originally contained 9503 features, reducing the feature set to between 9 and 27 did not compromise accuracy, maintaining performance above 99%. Iqubal et al. (2024) addressed the class imbalance problem by employing a hybrid approach that combined SMOTE-ENN for data balancing with PCA for dimensionality reduction. Likewise, studies such as Bhattacharya and Goswami (2018) and Mahindru et al. (2024) have shown that feature selection not only preserves detection performance but also significantly enhances computational efficiency.

Beyond numerical feature-based methods, classification using image processing techniques has emerged as an innovative direction. Falana et al. (2022) converted malware samples into RGB images and enriched the dataset using Deep Generative Adversarial Networks (DGAN), achieving high performance with CNN-based models. Rahman et al. (2024) advanced this approach further by employing the Vision Transformer (ViT) architecture, reaching an impressive accuracy of 99.94% on the Malimg dataset. Brown et al. (2024) highlighted the importance of optimization processes by utilizing automated systems such as AutoML.

In this study, feature selection was applied to the TUANDROMD dataset. Using the SelectKBest method based on ANOVA F-values, the ten most discriminative features were identified, and a Random Forest classifier was trained with this reduced feature set. Through this approach, the impact of dimensionality reduction on classification performance was evaluated.

Material and Methods

Dataset

In this study, the TUANDROMD (Tezpur University Android Malware Dataset) developed by Borah et al. (2020) and openly available in the UCI Machine Learning Repository was utilized. The dataset comprises 4465 samples (2415 malware + 2050 benign) and 241 features.

The label column is denoted as “Label,” where 0 represents benign and 1 represents malware. Missing values were checked, and observations with incomplete entries in the target column were removed to ensure data integrity.

Data Preprocessing

Since missing values were found only in the target variable, those rows were excluded. As no missing values were present in the independent variables, no additional imputation techniques were required. The dataset was split into training (80%) and testing (20%) subsets using stratified sampling (random_state=42). Stratified selection is a widely recommended approach to preserve model performance in the presence of imbalanced class distributions (Pedregosa et al., 2011).

Scaling and Feature Selection

Although the Random Forest algorithm is not sensitive to scaling, a StandardScaler was incorporated into the pipeline. This step is particularly necessary for variance-based methods such as SelectKBest to yield accurate results (Han et al., 2012). Feature selection was performed using the SelectKBest method with the ANOVA F-test (*f_classif*). This approach evaluates the degree of independence between each feature and the target class, selecting the 10 most significant features. ANOVA F-test is widely employed to identify features that best explain variance differences across classes (Guyon and Elisseeff, 2003).

Model Training

The Random Forest Classifier was employed for model training. Owing to its random subsampling and ensemble of multiple decision trees, Random Forest is a robust method that mitigates overfitting in high-dimensional datasets (Breiman, 2001). The training process was executed automatically within the pipeline, incorporating scaling and feature selection.

Hyperparameter Optimization

To enhance model performance, GridSearchCV was applied. A three-fold cross-validation (cv=3) was conducted to search across the following parameters:

n_estimators: [100, 300]

max_depth: [None, 10, 20]

min_samples_split: [2, 5]

Grid Search systematically explores a predefined parameter space to identify the combination yielding the highest accuracy (Bergstra and Bengio, 2012). The optimal parameters identified were automatically stored in the best_estimator_ attribute.

Experimental Study

A classification report was generated on the test data, including precision, recall, F1-score, and accuracy metrics. These measures provide a detailed evaluation of the model's performance across each class (Sokolova and Lapalme, 2009).

Confusion Matrix

The difficulty of distinguishing between classes was evaluated by visualizing which classes were more easily confused. The confusion matrix model is used as an important tool in understanding error typology (Sokolova and Lapalme, 2009).

Result and Discussion

In this study, a Random Forest-based classification model was developed on the TUANDROMD dataset using a transparent, reproducible, and minimal pipeline. The designed pipeline was structured to ensure the traceability and reproducibility of each stage of the data flow. Within this framework, modern machine learning techniques such as data preprocessing, feature selection, and hyperparameter optimization were systematically and holistically applied.

The classification performance of the model was comprehensively evaluated using widely accepted metrics, including accuracy, sensitivity, specificity, and F1-score. Furthermore, the obtained performance results were compared across different data partitions to analyze the stability and generalizability of the model.

The model's performance on the test set was presented in Table 1. Overall, the model achieved a high accuracy of 97%. For the majority class (class 1.0), precision, recall, and F1-score values were 0.98, indicating excellent performance. For the minority class (class 0.0), precision was 0.94, recall 0.91, and F1-score 0.92. These results demonstrate that the model effectively distinguishes both majority and minority classes. Macro-average and weighted-average metrics ranged between 95% and 97%, confirming that the model maintained balanced performance despite class imbalance.

Table 1. Classification report on the test dataset.

	Precision	Recall	F1-Score	Support
Class 0.0	0.94	0.91	0.92	180
Class 1.0	0.98	0.98	0.98	713
Accuracy			0.97	893
Macro-average	0.96	0.95	0.95	893
Weighted-average	0.97	0.97	0.97	893

According to the confusion matrix results presented in Figure 1, out of a total of 180 benign (Goodware) samples, 163 were correctly classified as Goodware by the model, while 17 were mistakenly assigned to the Malware class. Similarly, among 713 malicious (Malware) samples, 702 were accurately predicted as Malware, with only 11 misclassified as Goodware. This distribution indicates that the model exhibits low error rates for both classes. In particular, the high accuracy achieved in detecting Malware samples, which constitute the majority class in the dataset, demonstrates the model's strong capacity to distinguish harmful behaviors. Conversely, the classification performance observed for the minority class, Goodware, remains at an acceptable level despite class imbalance, suggesting that the model does not exhibit excessive bias toward a single class.

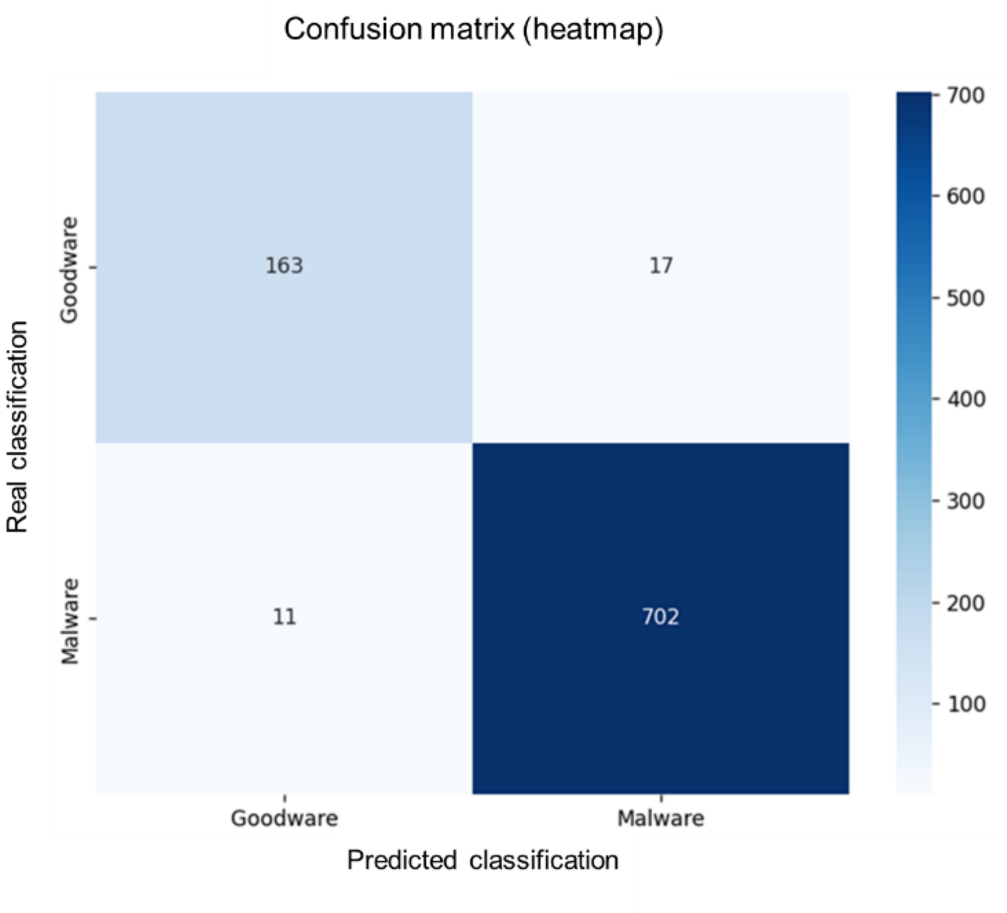

Figure 1. Confusion matrix (heatmap) on the test dataset. Values along the diagonal represent correct predictions, whereas off-diagonal values indicate misclassifications.

Effect of Data Preprocessing Steps

Data preprocessing steps have been one of the fundamental determinants of model performance. The removal of missing values, the stratified train-test split, and the scaling procedures contributed to the model's consistent and balanced performance.

Feature Selection and Model Performance

During the feature selection stage, the SelectKBest method identified the 10 most significant attributes in the dataset. This procedure reduced the risk of overfitting, lowered computational cost, and improved classification accuracy. Compared with models trained without feature selection, this approach provided a more balanced and stable performance.

Contribution of Hyperparameter Optimization

The hyperparameter search conducted with GridSearchCV enabled the structural parameters of the Random Forest algorithm to be adjusted in accordance with the dataset. Through this process, the number of trees (*n_estimators*), the maximum depth (*max_depth*), and the minimum number of samples required for splitting (*min_samples_split*) were optimized, thereby improving model accuracy. These findings confirm that hyperparameter tuning has a significant impact on the performance of tree-based methods.

This study performed the search on a predefined set of hyperparameters. However, with alternative approaches, larger and richer parameter spaces can be explored. Future research may incorporate different feature selection techniques (e.g., Recursive Feature Elimination, Genetic Algorithms, Permutation Importance), faster ensemble learning algorithms such as XGBoost or LightGBM, and interpretability analyses of the selected features to identify which permissions or API calls are most discriminative.

In conclusion, this study has demonstrated the significant impact of constructing a well-designed machine learning pipeline on classification performance. By systematically applying data preprocessing steps, employing effective feature selection, and tuning hyperparameters appropriately, the Random Forest model achieved high accuracy and balanced performance on the TUANDROMD dataset. These findings highlight that tree-based classification methods, when combined with systematic data preparation processes, serve as highly effective analytical tools.

Conflict of Interest

The author declares that for this article there are no actual, potential or perceived conflict of interest.

Author Contributions

N.D. performed all the experiments and drafted the main manuscript text. The author reviewed and approved the final version of the manuscript.

Ethical Approval Statements

No ethics committee permissions are required for this study.

Data Availability

The data used in the present study are available upon request from the corresponding author.

References

Bensaoud, A., Kalita, J., Bensaoud, M. (2024). A survey of malware detection using deep learning. *Machine Learning With Applications*, 16, 100546.

Bergstra, J., Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13, 281-305.

Bhattacharya, A., Goswami, R.T. (2018). Community based feature selection method for detection of android malware. *Journal of Global Information Management*, 26(3), 54-77.

Borah, P., Bhattacharyya, D.K., Kalita, J.K. (2020). Malware dataset generation and evaluation. In: Book of Proceedings. 2020 IEEE 4th Conference on Information & Communication Technology (CICT), 03-05 December 2020, Chennai, India, pp. 1-6.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5-32.

Brown, A., Gupta, M., Abdelsalam, M. (2024). Automated machine learning for deep learning based malware detection. *Computers & Security*, 137, 103582.

Falana, O.J., Sodiya, A.S., Onashoga, S.A., Badmus, B.S. (2022). Mal-Detect: An intelligent visualization approach for malware detection. *Journal of King Saud University-Computer and Information Sciences*, 34(5), 1968-1983.

Guyon, I., Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3, 1157-1182.

Han, J., Kamber, M., Pei, J. (2012). Data mining: concepts and techniques. Morgan Kaufmann.

Iqubal, A., Tiwari, S.K., Azad, S., Paswan, M.K. (2024). Android based malware detection technique using machine learning algorithms. In: Book of Proceedings. 2024 First International Conference on Pioneering Developments in Computer Science & Digital Technologies (IC2SDT), 02-04 August 2024, Delhi, India, pp. 610-615.

Mahindru, A., Arora, H., Kumar, A., Gupta, S.K., Mahajan, S., Kadry, S., Kim, J. (2024). PermDroid a framework developed using proposed feature selection approach and machine learning techniques for Android malware detection. *Scientific Reports*, 14(1), 10724.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E. (2011). Scikit-learn: machine learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.

Polatidis, N., Kapetanakis, S., Trovati, M., Korkontzelos, I., Manolopoulos, Y. (2024). FSSDroid: Feature subset selection for Android malware detection. *World Wide Web*, 27(5), 50.

Rahman, M.M., Hossain, M.D., Ochiai, H., Kadobayashi, Y., Sakib, T., Ramadan, S.T.Y. (2024). Vision Based Malware Classification Using Deep Neural Network with Hybrid Data Augmentation. In: Book of Proceedings. 10th International Conference on Information Systems Security and Privacy (ICISSP), 26-28 February 2024, Rome, Italy, pp. 823-830.

Smmarwar, S.K., Gupta, G.P., Kumar, S. (2024). Android malware detection and identification frameworks by leveraging the machine and deep learning techniques: A comprehensive review. *Telematics and Informatics Reports*, 14, 100130.

Sokolova, M., Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4), 427-437.

Wajahat, A., He, J., Zhu, N., Mahmood, T., Saba, T., Khan, A.R., Alamri, F.S. (2024). Outsmarting Android Malware with Cutting-Edge Feature Engineering and Machine Learning Techniques. *Computers, Materials & Continua*, 79(1). 651-673.