International Journal of Science and Research Archive

eISSN: 2582-8185

Cross Ref DOI: 10.30574/ijsra

Journal homepage: https://ijsra.net/

(RESEARCH ARTICLE)

Check for updates

Classifying android malware to secure mobile platforms using machine learning and deep learning approaches

Sayed Anower * and Taofica Amrine

Department of Computer Science and Engineering, Port City International University, Chittagong-4212, Bangladesh.

International Journal of Science and Research Archive, 2026, 18(02), 303–310

Publication history: Received on 29 December 2025; revised on 03 February 2026; accepted on 06 February 2026

Article DOI: https://doi.org/10.30574/ijsra.2026.18.2.0229

Abstract

The widespread use of Android devices has made them a prime target for cyberattacks, leading to an alarming rise in mobile malware. Detecting and classifying such threats effectively is essential to securing user data and maintaining platform integrity. This study presents a comprehensive framework for Android malware classification using a combination of machine learning (ML), deep learning (DL), and hybrid ensemble approaches. The research utilizes the CICMalDroid2020 dataset, containing 11,598 Android applications categorized into five classes: Adware, Banking Malware, SMS Malware, Riskware, and Benign. The data underwent extensive preprocessing, including normalization, label encoding, balancing, and feature extraction using the LASSO technique. Various ML algorithms such as Random Forest (RF), Decision Tree (DT), K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Extreme Gradient Boosting (XGB) were implemented alongside DL models like ANN, CNN, RNN, and LSTM. Additionally, hybrid models combining CNN with LSTM, XGB, and RF were developed to enhance detection accuracy. Experimental results indicate that the proposed models outperform existing approaches in malware classification. The XGB model achieved the highest accuracy of 95.74%, followed by the Ensemble (RF, ET, XGB) with 95.44%, surpassing previous work. These results demonstrate that integrating ensemble and deep learning architectures provides improved generalization, robustness, and precision in detecting Android malware.

Keywords: Android Malware; Riskware; Machine Learning; Deep Learning

1. Introduction

The past ten years have seen the mobile devices evolve to become not just a mere communication device, but also a part of life that enables banking, healthcare, and connectivity to the world. Android operating system is the main system that is being used in the smartphone market in the world because of its open-source architecture and flexibility. Nonetheless, this has led to it becoming a major target of cybercriminals. Android malware has led to exponential increase of user privacy, financial security and integrity of mobile ecosystems. Recent cybersecurity reports note that millions of new malware samples have been identified each year with advanced variants using code obfuscation, dynamic loading and polymorphic patterns in order to avoid detection.

Conventional defense systems, which include signature-based and heuristic scanning, are becoming more and more ineffective against the new modern threats. These standardization approaches depend on established trends and therefore are not aware of the zero-day attacks and mutated malware strains. It is therefore of critical importance that intelligent and adaptive systems are developed that can generalize detection logic to unprecedented threats. New developments in the field of Machine Learning (ML) and Deep Learning (DL) provide a potential path to the automation of malware classification based on the acquisition of specific patterns based on high-dimensional data.

* Corresponding author: Sayed Anower

There are still critical issues that are evident despite the promise of AI-driven security. To begin with, it is complicated to extract features with encryption and packing methods. Second, current ML-based systems frequently have difficulties with large dimensionality of Android Application Package (APK) features that causes computational bottlenecks. Third, Deep Learning models are highly accurate, but usually black-box and therefore not interpretable, or computationally expensive enough to be used in a mobile environment. As such, building a detection system with good accuracy, flexibility, and computational performance is an open research problem.

In order to deal with these issues, this paper suggests a detailed system of Android malware classification based on the combination of the Machine Learning, Deep Learning, and Hybrid Ensemble methods. This dataset is the CICMalDroid2020 which is a powerful benchmark with 11,598 samples in 5 different categories: Adware, Banking Malware, SMS Malware, Riskware, and Benign applications. In contrast to the classical studies where many studies usually utilize raw and high-dimensional features, we introduce an LASSO (Least Absolute Shrinkage and Selection Operator) feature extraction and dimensionality reduction method of features to the analysis. We use superior architectures such as Hybrid CNN-LSTM models, and optimized Extreme Gradient Boosting (XGB) to distinguish between benign and malicious applications with a high level of accuracy.

The major findings of this paper can be summarized as follows:

- **Strong Classification Framework**: Our model is built with an extreme gradient boosting (XGB) and Hybrid Ensemble on the strategy. With the high-dimensionality feature selection using LASSO, we solve the problem of high dimensionality of the Android malware features, with a maximum accuracy of 95.74 percent that is higher in comparison with typical baseline techniques.
- **Comprehensive Comparative Analysis**: We present a substantive benchmark of different algorithms, including both standard ML classifiers (Random Forest, SVM, KNN) and deep neural networks (ANN, CNN, RNN, LSTM) and combinations of them (CNN-LSTM, CNN-RF) and analyze their suitability in tracking the evolving malware variants.
- **Multi-Class Threat Detection**: We identify the key Android application behaviors in particular Malware families; Adware, Banking Malware, SMS Malware, and Riskware; which will provide us with a detailed understanding of the unique features of the newest mobile threats.
- **Security Enhancement:** The proposed model will be a step towards the next-generation mobile threat defense, which can be scalable and introduces both interpretability of the ensemble approach and feature-learning that is provided by the deep learning.

The rest of this paper will be structured as follows: Section 2 will conduct a review of related work in Android malware detection. Section 3 outlines the suggested methodology, preprocessing procedures, and features extraction procedures. Section 4 includes the description of the experiment setup, metrics of the performances, and analysis of the results. Lastly, Section 5 concludes the paper and provides directions in the future research.

2. Literature Review

Recent cybersecurity studies have found the use of Machine Learning (ML) and Deep Learning (DL) to classify Android malware to be a particularly attractive subject. The available literature can be broadly categorized into three groups: feature optimization measures, high-end neural architecture, and domain threat detection. The section refutes the contribution of the proposed study by reviewing the major contributions in the fields and pinpointing the limitations.

2.1. Feature Selection and Ensemble Approaches

The major malware detection issues relate to the high dimensionality of sets of features that may cause overfitting and computational inefficiency. This has been alleviated by a number of studies suggesting advanced feature selection techniques. As an example, [1] proposed a feature selection approach used in the form of gain ratio in combination with an ensemble learning model. This method was tested on the CICMalDroid2020 dataset and was able to significantly improve the performance of single classifiers, and their combination of Random Forest, Extra Trees, and k-NN gave an accuracy of 94.57%. This paper confirms the effectiveness of the feature space reduction in order to increase the classification accuracy.

On the same note, [2] also suggested Adaptive Feature Selection (AFS) strategy to API call sequences. They were able to ablate and show that AFS shows significant improvement in F1-scores and complexity reduction which is confirmed by strong metrics such as the Matthews Correlation Coefficient (MCC) by intelligently shortening long sequences in order to concentrate on the data relevant. Similarly, [3] has used Multi-Objective Genetic Algorithms (MOGAs) to adopt

important behavioral attributes. Their hybrid structure also narrowed down the feature count (335) to 200 and displayed 93.33% accuracy with J48 and REP Tree classifiers, which demonstrates the new engine of evolutionary algorithms over the old one in classical selection.

2.2. Deep Learning and Hybrid Architectures

Although ML is interpretable, Deep Learning has been demonstrated to perform better in pattern recognition. [4] suggested a hybrid CNN-LSTM state-of-the-art architecture which processes API calls, opcodes as N-grams. Their model had astonishing high-accuracy of 99.91, surpassing high-quality models, such as Vision Transformers (Swin-T), which indicates that more simple, well-tuned hybrid models can work miraculously.

Considering pure ML scalability, [5] tested different models using API sequences obtained out of nine different malware families. Their findings found XGBoost to be the best classifier with 98.87 percent accuracy that supports the possibility of gradient boosting frameworks in sequential data. In addition, [6] also proposed a statistical method based on L-moments of Abstract Syntax Trees (ASTs). This approach offered a profound structural insight into code with better detection of new malware variants as compared to conventional analysis of the code.

2.3. Specialized Threat Detection and Surveys

The cybersecurity environment is heterogeneous and needs dedicated strategies to address various sources of threat. The article [7] offered an extensive literature review on the application of ML in automated defense processes, focusing on how unsupervised learning can be used to identify zero-day threats and how federated learning can be used to build collective resilience.

The targeted research has been made in specific areas, including the ransomware and IoT domain. Triple-Layer Bayesian Euclidean Curve Algorithm was introduced to conduct ransomware classification as proposed by [8]. Probabilistic approach can well deal with the obfuscated code and be better recalled as compared to traditional SVM models used in the generalization. Within the cryptocurrency field, [9] adopted a semi-supervised ensemble model to the BitcoinHeist data, which worked well to detect known and unknown ransomware in the blockchain dealings. Lastly, a solution to resource limitation, [10] introduced a lightweight framework of detecting IoT malware based on Matrix Block Mean Downsampling (MBMD). This method compressed the trained features without loss of local means with an F1-score of more than 99% with a low level of computing cost.

2.4. Summary and Gap Analysis

The evaluated literature shows that although the features selection (e.g. gain ratio, MOGA) and the hybrid models (e.g. CNN-LSTM) have reported high accuracy, there are still issues of balancing between the computational and multi-class classification. There are numerous high-accuracy models that can be considered black boxes or that demand large amounts of resources, which are not suitable in real-time implementation. More so, although [1] used CICMalDroid2020, they reached their highest accuracy of 94.57%. The research undertaken will fill these gaps by presenting a framework of LASSO-based feature selection combined with the XGBoost and Hybrid CNN-LSTM models, with the goal of achieving higher results compared to current benchmarks on the same dataset and guarantee the robustness of the models and their generalization.

3. Methodology

3.1. Proposed Framework Overview

This paper presents a multi-level, end-to-end Android malware detection framework, which integrates the best feature selection with deep learning hybrid architectures. The methodology had five main steps, as shown in the figure 1: (1) Data Collection using the CICMalDroid2020 repository; (2) Preprocessing involving data cleaning, encoding, normalization, and balancing; (3) Feature Extraction with the use of the LASSO (Least Absolute Shrinkage and Selection Operator) algorithm, which is used to reduce dimensionality; (4) Model Training that is carried out in three different architectures: (3) Machine Learning (ML), (3) Deep Learning (DL), and (4) Hybrid architect.

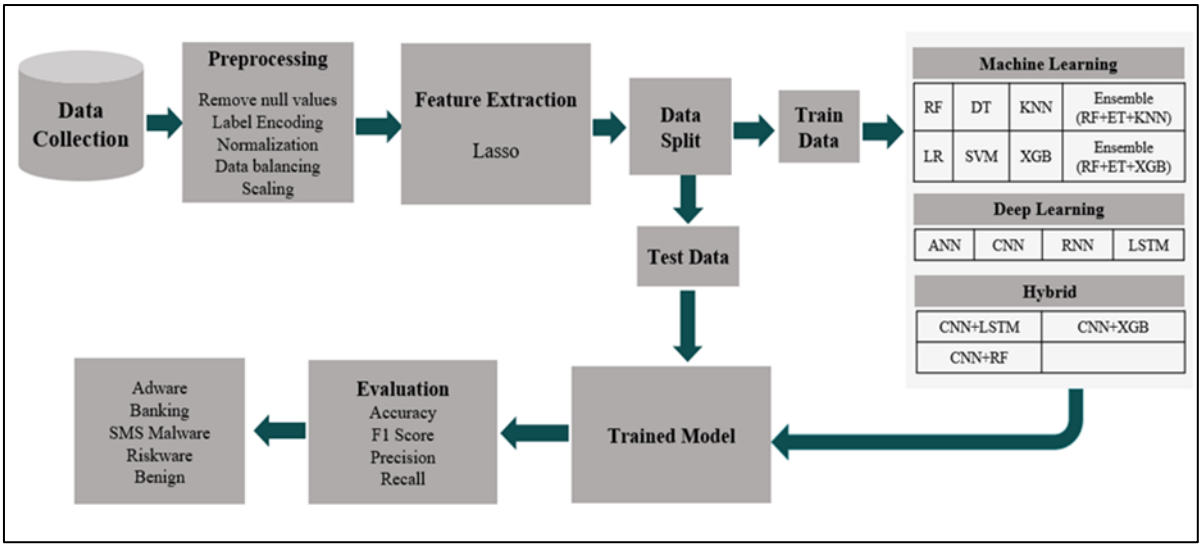

Figure 1 Methodology of This Study

3.2. Dataset Description

The experimental assessment involves the use of the CICMalDroid2020 dataset that was created by the Canadian Institute for Cybersecurity (CIC) [11]. This dataset can be used as a reference in the study of mobile security because it is concentrated on obfuscated malware, which is a burning issue in Android security nowadays. The sample size is 11,598 Android application samples, which can be grouped in five different classes:

- **Adware**: The application developed to display invasive advertisements or to earn money by the means of false ad-clicks.
- **Banking Malware**: malicious software that has been specially designed to steal financial credentials, to intercept two-factor authentication tokens and to compromise banking transactions.
- **SMS Malware**: A software that uses the SMS services to deliver premium-rate messages or to intercept personal communications without the consent of the user.
- **Riskware**: legal applications that are potentially vulnerable to security risks as they contain vulnerabilities that can be exploited maliciously.
- **Benign**: Android applications that do not display any signs of maliciousness or criminal intent.

3.3. Data Preprocessing and Transformation

In order to get the quality of input data and enhance convergence of model, raw data was processed through a strict preprocessing pipeline:

- **Data Cleaning:** Null and missing values were eliminated to avoid errors during the calculations process.
- **Label Encoding**: Encoding of categorical labels (e.g. malware families) was translated into numerical values, which can be used by machine learning algorithms.
- **Normalization and Scaling**: Feature values were brought to the normal range to avoid the domination on the loss of features with bigger magnitudes, especially in distance-based algorithms such as KNN and gradient-based models such as CNNs.
- **Data Balancing**: To overcome the class imbalance of malware datasets, data balancing methods were used to make sure the model is not biased towards one of the dominant classes (Benign).

3.4. Feature Selection using LASSO

Android Application Package (APK) features are high-dimensional, and, in this case, LASSO (Least Absolute Shrinkage and Selection Operator) has been applied to extract features. LASSO is an L1-regularization method which punishes the magnitude of coefficients. In contrast to Ridge regression, LASSO has the ability of shrinking coefficients to zero, which is effectively feature selection, removing irrelevant/redundant features. This will play an essential role in mitigating the complexity of computations and increasing the generalization property of the hybrid models.

3.5. Classification Models

The framework evaluates three categories of classifiers to establish a comprehensive benchmark:

3.5.1. Machine Learning and Ensembles

Among the commonly used classifiers that we used are Random Forest (RF), Decision Tree (DT), K-Nearest Neighbor (KNN), Logistic Regression (LR), Support Vector Machine (SVM), and Extreme Gradient Boosting (XGB). We also constructed some particular Ensemble Voting Classifiers in order to combine predictions:

- Ensemble A: Random Forest + Extra Trees + KNN.
- Ensemble B: Random Forest + Extra Trees + XGB.

3.5.2. Deep Learning and Hybrid Architectures

In order to fit non-linear trends, we used Deep Learning models: Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM). Moreover, to make the best use of both the feature removal capabilities of CNNs and sequential learning or decision-making capabilities of other models we proposed Hybrid Models:

- **CNN + LSTM:** Combines spatial feature extraction with temporal sequence learning.
- **CNN + XGB:** Uses CNN for feature learning and XGBoost for high-performance classification.
- **CNN + RF:** Integrates deep features with the robustness of Random Forest.

3.6. Experimental Setup and Data Splitting

Data set was split into training and testing sets to avoid data leakage and to guarantee the objectivity of the evaluation. We performed a pilot ablation experiment with the aim to identify the best split ratio. We tested the accuracy of the model at the ratio of 70:30, 80:20 and 90:10 as shown in Table 1.

Table 1 Performance comparison of different data split ratios

Split Ratio (Train : Test)	Accuracy	Justification
70:30	0.9406	Standard split, but underutilized training data.
80:20	0.9462	Improved accuracy over 70:30.
90:10	0.9492	Selected Ratio. Achieved the highest accuracy, maximizing the data available for model learning while retaining a sufficient representative set for evaluation.

As a result, the data was divided into 90% Training Data on which models are being learned and 10% Testing Data on which models are being tested.

3.7. Evaluation Metrics

The standard classification metrics evaluated to test the performance of the proposed models are Accuracy, F1-Score, Precision, and Recall. These measures give the overall picture of the capability of the model to minimize False Positives (benign apps getting classified as malware) and False Negatives (malware being missed).

4. Results and Discussion

4.1. Experimental Results Overview

The suggested model was tested on the testing subset (10%) of the CICMalDroid2020 data. Our comparative analysis has been performed across three model categories, namely, Machine Learning (ML), Deep Learning (DL), and Hybrid architectures. Accuracy, Precision, Recall and F1-Score were the metrics used to determine the performance of each model. Table 2 shows the performance metrics of all the implemented algorithms in detail.

Table 2 Performance Comparison of Classifiers

Category	Model	Accuracy	Precision	Recall	F1-Score
Machine Learning	Random Forest (RF)	0.95	0.95	0.95	0.95
	Decision Tree (DT)	0.90	0.90	0.90	0.90
	K-Nearest Neighbor (KNN)	0.90	0.90	0.90	0.90
	XGBoost (XGB)	0.96	0.96	0.96	0.96
	Logistic Regression (LR)	0.81	0.81	0.81	0.81
	SVM	0.77	0.79	0.77	0.77
Ensemble	RF + ET + KNN	0.95	0.95	0.95	0.95
	RF + ET + XGB	0.95	0.95	0.95	0.95
Deep Learning	ANN	0.89	0.89	0.89	0.89
	CNN	0.91	0.91	0.91	0.91
	RNN	0.88	0.88	0.88	0.88
	LSTM	0.80	0.80	0.80	0.80
Hybrid	CNN + LSTM	0.85	0.85	0.85	0.85
	CNN + XGB	0.95	0.95	0.95	0.95
	CNN + RF	0.95	0.95	0.95	0.95

4.2. Performance Analysis

4.2.1. Machine Learning Classifiers

XGBoost (XGB) was the best-performing traditional ML algorithm with an accuracy of 95.74% (in Table 2, it was rounded to 0.96). This advantage can be explained by the fact that XGBoost has a gradient boosting structure that is highly effective in dealing with non-linear relationships and feature interactions which malware behavior exhibits. Random Forest (RF) was also able to perform well with 95 percent accuracy.

On the other hand Support Vector Machine (SVM) and Logistic Regression (LR) gave the lowest accuracy of 77 percent and 81 percent respectively. This finding is indicative that the decision boundary between malware families in the CICMalDroid2020 data is very non-linear, and linear classifiers cannot be effective without a complex map of the actual relationship between the kernel and its mapping.

4.2.2. Deep Learning and Hybrid Models

The Convolutional Neural Network (CNN) performed the highest standalone score (91%), in the Deep Learning category. Nonetheless, models based on sequence such as LSTM performed poorly (80%), which implies that simple patterns of the sequence (without spatial feature extraction) might not be effective in this feature set.

An important discovery is the effectiveness of Hybrid Models. The CNN + XGB and CNN + RF gave 95 percent accuracy, which is equal to the highest ensemble techniques. This confirms our hypothesis, that CNN, as a feature extractor (to extract local data correlations) and a strong tree-based classifier (such as XGBoost) gives better results than deep learning classifiers (Softmax) when used by itself.

4.3. Confusion Matrix Analysis

To further analyze the classification performance of the best-performing model (XGBoost), we generated the confusion matrix shown in Figure 2.

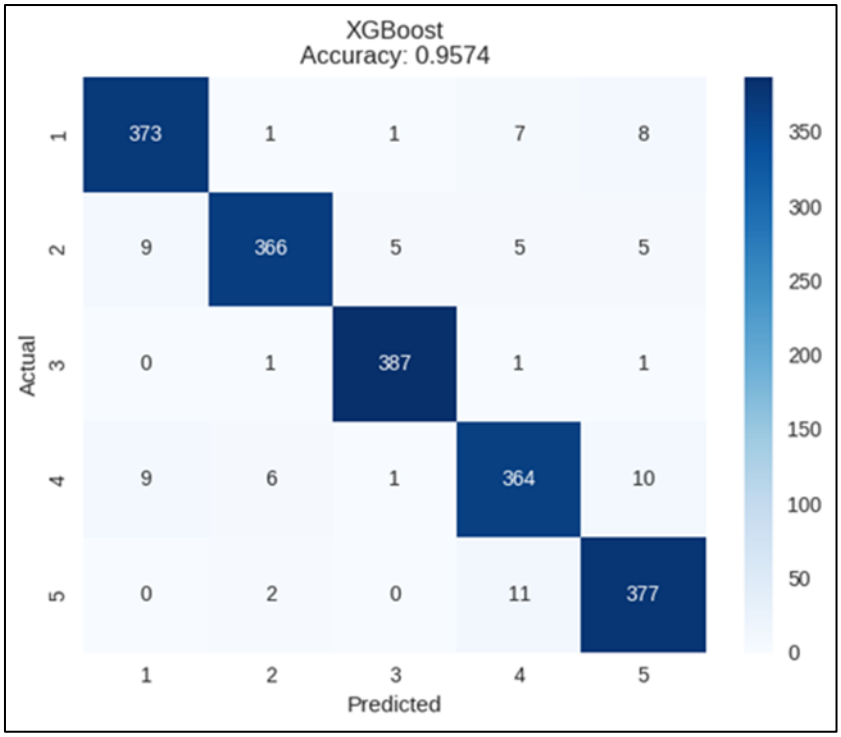

Figure 2 Confusion Matrix of XGB Model

The matrix presents the high values of the diagonal (True Positives) which implies that the model fits most of the samples of all of the five classes.

The small amount of the off-diagonal elements (misclassifications) proves the strength of the model. Confusion is the greatest between Class 1 and Class 5, indicating a possible shared behavioral characteristic between the two particular malware families (e.g. Adware and Riskware frequently have permission patterns in common). In general, it can be observed that XGBoost is a highly trustworthy multi-class Android malware classifier.

5. Conclusion

The malicious development of Android malware is increasing exponentially and hence requires stringent, automated detection systems that transcend the conventional signature-based systems. This research provided a multi-class Android malware detection comparative framework based on the CICMalDroid2020 dataset. Our combination of LASSO-based feature selection with an extensive range of Machine Learning, Deep Learning, and Hybrid architectures overcame both of these crucial challenges of high dimensionality and class imbalance.

According to our experimental findings, Deep Learning models such as CNNs have high feature extraction abilities (91% accuracy); nevertheless, the best performance was achieved by the traditional ensemble algorithms, i.e. Extreme Gradient Boosting (XGBoost), at 95.74% accuracy. This observation is important to note that, on tabular-style features derived out of APKs (such as permissions and API calls), tree-based gradient boosting models can be more effective than more complex recurrent models (such as LSTM, which only attained 80%). Moreover, the high accuracy of our Hybrid CNN+XGB model (95) is a guarantee that a deep feature extraction with effective classification heads can be a useful approach to detecting malware with high precision.

Finally the proposed framework offers a scalable and high-accuracy proposal that is capable of differentiating between fine-grained malware groups (Adware, Banking, SMS, Riskware) and low false positives, which will play an important part in making mobile platforms secure.

Future Scope

Although this paper provides a solid foundation to the classification of malware, a number of research opportunities can be identified in the future:

- **Adversarial Robustness**: The next generation will test the ability of the model to be resistant to adversarial attacks, where malware authors deliberately add perturbations so as to evade the detection. It will be essential to test the framework on the generated adversarial samples to determine its reliability in the real world.
- **On-Device Deployment**: We also intend to optimize the proposed lightweight models (namely the XGBoost and random forest models) to run on the resource-constrained Android devices, to provide real-time and offline security.
- **Explainable AI (XAI)**: To make hybrid models more transparent, we will incorporate the XAI methods such as SHAP (SHapley Additive exPlanations) or LIME. This will enable the security analysts to understand which API calls or permissions contributed to the specific malware classification.
- **Dynamic Analysis Integration**: The framework is currently based on the static characteristics. Subsequent releases will add dynamic analysis capabilities (e.g. system calls monitoring at runtime) to detect complex types of malware that applies standard dynamic code loading or packing algorithms to conceal their static signature.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] Ansori, D. B., Slamet, J., Ghufron, M. Z., Putra, M. A. R., & Ahmad, T. (2024). Android malware classification using gain ratio and ensembled machine learning. International Journal of Safety and Security Engineering, 14(1), 259-266.

[2] Njeri, N., Ivanov, O., Rodriguez, S., Richardson, A., & Delgado, C. (2024). Triple-layer bayesian euclidean curve algorithm for automated ransomware classification.

[3] Panda, B., Bisoyi, S. S., Panigrahy, S., & Mohanty, P. (2025). Machine learning techniques for imbalanced multiclass malware classification through adaptive feature selection. PeerJ Computer Science, 11, e2752.

[4] Sharma, B. P. (2024). Machine learning-driven approaches for contemporary cybersecurity: From intrusion detection and malware classification to intelligent incident response. Nuvern Machine Learning Reviews, 1(1), 22-32.

[5] Rose, A. J., Schubert Kabban, C. M., Graham, S. R., Henry, W. C., & Rondeau, C. M. (2024). Malware Classification through Abstract Syntax Trees and L-moments. Computers & Security (ISSN 0167-4048).

[6] Dib, Omar, Zhenghan Nan, and Jinkua Liu. "Machine learning-based ransomware classification of Bitcoin transactions." Journal of King Saud University-Computer and Information Sciences 36.1 (2024): 101925.

[7] Farfoura, M. E., Mashal, I., Alkhatib, A., Batyha, R. M., & Rosiyadi, D. (2025). A novel lightweight Machine Learning framework for IoT malware classification based on matrix block mean Downsampling. Ain Shams Engineering Journal, 16(1), 103205.

[8] Kale, G., Bostancı, G. E., & Celebi, F. V. (2024). Evolutionary feature selection for machine learning based malware classification. Engineering Science and Technology, an International Journal, 56, 101762.

[9] Bensaoud, A., & Kalita, J. (2024). CNN-LSTM and transfer learning models for malware classification based on opcodes and API calls. Knowledge-Based Systems, 290, 111543.

[10] Li, Yao, et al. "Meta-learning for multi-family android malware classification." ACM Transactions on Software Engineering and Methodology 33.7 (2024): 1-27.https://www.unb.ca/cic/datasets/maldroid-2020.html