Contents lists available at IdNNS
Indonesian Journal of Cybersecurity and Emerging Risks
Volume 1 ; Issue 1
journal homepage: https://journal.idnns.org/index.php/ijcyber

Ransomware Detection Using Light Gradient Boosting Machine Classifier

Abdullah Said Alkalbani[a,*], Deni Mulyana[b]

[a]Deputi Vice Chancellor for Academic Affair and Innovation, University of Buraimi, Oman
[b]Information Master Study Program, Nusa Putra University, Sukabumi, Indonesia

ARTICLE INFO

Keywords:
ransomware detection
LightGBM
multiclass classification
SMOTEENN

ABSTRACT

Ransomware continues to represent a critical cybersecurity threat across various sectors, leading to significant financial and operational disruptions. Traditional detection techniques, such as signature-based and behavior-based approaches, often fall short in identifying advanced ransomware variants that utilize polymorphism, obfuscation, and zero-day exploits. To overcome these limitations, this study proposes a machine learning-based framework for multi-class ransomware detection using the Light Gradient Boosting Machine (LightGBM) classifier. The model was trained and evaluated on a labeled Android dataset comprising diverse ransomware families alongside benign applications. To address the issue of class imbalance, the Synthetic Minority Oversampling Technique with Edited Nearest Neighbors (SMOTEENN) was employed, followed by thorough feature preprocessing and hyperparameter optimization. The optimized LightGBM model achieved an accuracy of 90.78% and a weighted F1-score of 90.73%, outperforming baseline classifiers such as Decision Tree and Random Forest. This study offers both theoretical and practical contributions by presenting a lightweight yet highly accurate detection system suitable for real-world deployment, and by validating the effectiveness of SMOTEENN when combined with LightGBM for handling imbalanced, multiclass cybersecurity datasets.

1. Introduction

Ransomware has become one of the most serious cybersecurity threats in the modern digital environment. This type of malware prevents users from accessing their data or systems by encrypting files or locking devices, then demands ransom payment for recovery. Ransomware attacks have affected many critical sectors, including healthcare, education, finance, and government institutions, causing financial losses, operational disruption, and risks to data privacy Coveware (2021); Stritch and Allyn (2021); Hagerty and Nygard (2024). The impact of ransomware is especially severe because attacks may interrupt essential services and create long recovery processes for organizations.

The growth of mobile technology has also increased the risk of ransomware attacks on Android devices. Android is widely used because of its open ecosystem and large application market, but these characteristics also make it attractive to malware developers. Android ransomware can perform malicious activities such as locking the screen, encrypting files, stealing sensitive data, and abusing permissions granted by users. Chen et al. Chen et al. (2018) showed that Android ransomware has diverse malicious behaviors and requires specific detection mechanisms to identify threats in real time.

Traditional ransomware detection methods commonly rely on signature-based and rule-based techniques. Although these methods are useful for detecting known malware, they are less effective against new ransomware variants that apply code obfuscation, polymorphism, metamorphism, and fileless attack strategies. Polymorphic malware can change its code structure to avoid detection, while fileless malware can execute malicious behavior without leaving conventional file-based artifacts Avhankar and Khandare (2025); Kara (2023). In addition, zero-day ransomware variants are difficult to detect because they may not match existing malware signatures or previously observed attack patterns Zahoora et al. (2022); Cen et al. (2024); Sangher et al. (2024).

To overcome these limitations, machine learning has been widely applied in malware and ransomware detection. Machine learning models can learn patterns from large datasets and classify malicious behavior based on features such as network traffic, memory traces, API call sequences, and application behavior. Previous studies have demonstrated the effectiveness of supervised machine learning for Android ransomware detection using traffic-based features

*Corresponding author
abdullah.s@uob.edu.om (A.S. Alkalbani); deni.mulyana@nusaputra.ac.id (D. Mulyana)
ORCID(s):

Albin Ahmed and Shaikh (2023), memory-based features Aljabri et al. (2024), and API call sequences Nguyen and Lee (2021). Comparative research has also shown that algorithms such as Random Forest, Support Vector Machine, Neural Network, and K-Nearest Neighbors can be used for Android ransomware classification, although their performance differs depending on the dataset and feature representation Momposhi et al. (2025); Imran et al. (2024); Kirubavathi and Deepa (2024).

Among various machine learning algorithms, Light Gradient Boosting Machine (LightGBM) has gained attention because of its efficiency, scalability, and strong performance on high-dimensional datasets. LightGBM uses gradient boosting with leaf-wise tree growth and histogram-based splitting, allowing it to train faster while maintaining high classification accuracy. Previous research has applied LightGBM for malware detection and ransomware detection, showing that it can achieve strong performance in identifying malicious samples Alkasassbeh and Abbadi (2020); Gao et al. (2022); Nguyen and Lee (2021); Ali et al. (2024). Recent studies also indicate that LightGBM can be combined with optimization or feature selection techniques to improve Android malware classification performance Zhou et al. (2024).

However, ransomware datasets often contain imbalanced class distributions, where some ransomware families have fewer samples than others. This imbalance can reduce classification performance because the model may become biased toward majority classes. Therefore, data balancing and feature selection are important steps in building an effective ransomware detection model. Previous research on ransomware early detection has emphasized the importance of evaluating models using metrics such as accuracy, precision, recall, and F1-score to ensure reliable detection performance AlHashmi et al. (2024); Hussain et al. (2025).

Based on these challenges, this study proposes a machine learning-based framework for multiclass ransomware detection using the LightGBM classifier. The study uses an Android ransomware dataset containing benign applications and several ransomware families. To improve detection performance, preprocessing, feature selection, and data balancing techniques are applied before model training. The performance of LightGBM is then compared with baseline classifiers such as Decision Tree and Random Forest. This study aims to provide a lightweight and accurate ransomware detection approach that can support real-world cybersecurity defense against Android ransomware threats.

2. Research Methodology

2.1. Research Methodology Process

This study applies a systematic research methodology to develop a multiclass ransomware detection framework using the Light Gradient Boosting Machine (LightGBM) classifier. The methodology is designed to ensure that the dataset is properly prepared, the model is trained effectively, and the evaluation results are reliable Madanayaka et al. (2023). The overall process consists of four main stages: data understanding, data preparation, model training, and model evaluation.

In the data understanding stage, this study uses an Android ransomware dataset containing benign applications and several ransomware families. The dataset provides network traffic and behavioral features that can be used to distinguish ransomware activity from normal application behavior. Similar Android ransomware detection studies have shown that traffic-based and behavior-based features are useful for identifying malicious Android applications Albin Ahmed and Shaikh (2023); Chen et al. (2018).

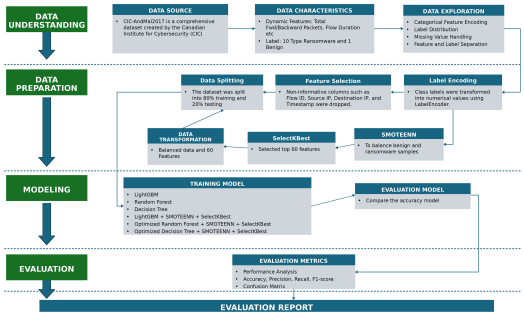

Figure 1: Research Methodology Process

The second stage is data preparation. In this stage, irrelevant features such as metadata columns are removed, missing values are handled, and non-numeric data are transformed into a suitable format for machine learning. Because ransomware datasets often have imbalanced class distributions, this study applies the Synthetic Minority Oversampling Technique with Edited Nearest Neighbors (SMOTEENN) to balance the dataset. Data balancing is important because an imbalanced dataset may cause the classifier to focus more on majority classes and reduce detection performance on minority ransomware families AlHashmi et al. (2024).

After preprocessing, feature selection is performed to reduce dimensionality and improve computational efficiency. This study uses SelectKBest with the ANOVA F-statistic to select the most relevant features for classification. Feature selection helps remove redundant or less informative features, allowing the model to focus on the most significant patterns related to ransomware behavior.

The third stage is model training. In this stage, the LightGBM classifier is trained using the prepared dataset. LightGBM is selected because it is efficient for large-scale and high-dimensional data and has shown strong performance in malware and ransomware detection tasks Alkasassbeh and Abbadi (2020); Gao et al. (2022); Nguyen and Lee (2021). For comparison, baseline classifiers such as Decision Tree and Random Forest are also trained to evaluate whether LightGBM provides better classification performance.

The final stage is model evaluation. The trained models are evaluated using accuracy, precision, recall, F1-score, and confusion matrix. These metrics are used to measure the model's ability to classify benign and ransomware samples correctly. The evaluation results are then compared to determine the most effective model for multiclass ransomware

detection. This research methodology is expected to produce a lightweight, accurate, and practical detection framework for Android ransomware.

2.2. Dataset Overview

The CIC-AndMal2017 dataset, curated by the Canadian Institute for Cybersecurity (CIC), is used as the benchmark dataset in this study. This dataset contains Android malware traffic data that can be used for machine learning-based malware detection. The dataset provides several feature sets that enable the development of reliable classification models for detecting Android malware, especially ransomware. Previous studies have also used Android traffic and behavioral features to distinguish ransomware activities from benign applications Albin Ahmed and Shaikh (2023); Chen et al. (2018).

Initially, the dataset contained 350,968 entries. After data cleansing, including the removal of duplicate and invalid records, the final dataset used in this study was reduced to 349,855 entries. The dataset consists of 85 columns, where 84 columns are used as input features and one column is used as the class label. The class label indicates whether a sample belongs to a benign application or one of several ransomware families. The dataset structure is summarized in Table 1.

Table 1: Sample of network flow records

Flow ID	Source IP	Destination IP	Protocol
203.205.158.63-10.42.0.151-80-43606-6	10.42.0.151	203.205.158.63	6
172.217.3.110-10.42.0.151-443-42470-17	10.42.0.151	172.217.3.110	17
203.205.158.60-10.42.0.151-443-37329-6	10.42.0.151	203.205.158.60	6
10.42.0.151-10.42.0.1-62877-53-17	10.42.0.151	10.42.0.1	17
10.42.0.151-123.125.29.138-47278-443-6	123.125.29.138	10.42.0.151	6
10.42.0.151-10.42.0.1-13621-53-17	10.42.0.151	10.42.0.1	17
10.42.0.151-10.42.0.1-24163-53-17	10.42.0.151	10.42.0.1	17
180.149.138.210-10.42.0.151-80-59003-6	180.149.138.210	10.42.0.151	6
157.55.56.174-10.42.0.151-40022-43477-17	10.42.0.151	157.55.56.174	17
10.42.0.151-64.71.142.96-34062-80-6	64.71.142.96	10.42.0.151	6

2.3. Model Training

This study employs the Light Gradient Boosting Machine (LightGBM) classifier as the main model for ransomware detection. LightGBM is an open-source gradient boosting framework designed for efficient and scalable machine learning. It is suitable for high-dimensional and large-scale datasets because it uses histogram-based decision tree learning and leaf-wise tree growth. These characteristics allow LightGBM to train faster while maintaining strong classification performance Alkasassbeh and Abbadi (2020); Gao et al. (2022); Kalhana et al. (2024).

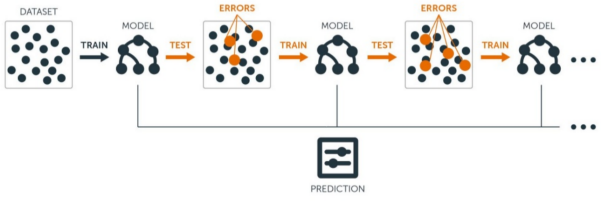

Figure 2: LightGBM Structure

LightGBM has been widely applied in malware and ransomware detection because it can handle complex feature patterns and produce accurate classification results. Nguyen and Lee Nguyen and Lee (2021) demonstrated that LightGBM can effectively detect ransomware using API call sequences, while Zhou et al. Zhou et al. (2024) showed that LightGBM can be combined with optimization techniques to improve Android malware classification. The general structure of LightGBM is illustrated in Fig. 2.

Gradient boosting works by combining multiple weak learners, usually decision trees, into a stronger predictive model. At each iteration, a new weak learner is added to correct the residual errors produced by the previous model. The prediction process can be represented as follows:

$$F_t(x) = F_{t-1}(x) + h_t(x) \tag{1}$$

where $F_t(x)$ represents the updated model prediction, $F_{t-1}(x)$ represents the previous model prediction, and $h_t(x)$ represents the weak learner added at the t-th iteration.

2.4. Model Implementation

The experimental procedure was structured under two distinct configurations: a baseline scenario, which utilized the raw dataset without any optimization techniques, and an optimized scenario, in which class imbalance was addressed using the SMOTEENN technique and dimensionality was reduced through feature selection using SelectKBest. This dual-configuration approach allows for a fair and comprehensive assessment of how preprocessing strategies influence model performance across a range of evaluation metrics.

The experimental design involves two modeling scenarios: baseline and optimized. In the baseline scenario, all three classifiers — LightGBM, Random Forest, and Decision Tree — are trained on the original dataset without applying any form of data balancing or feature selection. In contrast, the optimized scenario utilizes the same classifiers trained on the dataset that has been preprocessed using SMOTEENN and SelectKBest.

Within this framework, LightGBM is emphasized as the primary model due to its well-established effectiveness in handling large-scale and imbalanced datasets, as well as its superior performance in previous studies. Each model is configured with a set of hyperparameters to control its learning behavior. For the LightGBM model, the parameters used are as follows: `num_leaves = 320`, `learning_rate = 0.03`, `n_estimators = 800`, `max_depth = 20`, `min_child_samples = 10`, `subsample = 0.85`, `colsample_bytree = 0.85`, `reg_alpha = 0.2`, and `reg_lambda = 0.2`. For the Random Forest model, an ensemble of 100 decision trees is created using `n_estimators = 100`.

The models are trained using the `fit()` method on the training set `(X_train, y_train)`. Minor adjustments, such as tuning `max_depth` and increasing the number of estimators, are applied in the optimized configuration. The Decision Tree model follows a similar process. To ensure reproducibility and consistency across experiments, all models are implemented using a structured machine learning pipeline.

2.5. Model Evaluation

Evaluating the performance of machine learning models is essential to determine their effectiveness in solving classification problems such as ransomware detection. In this study, several evaluation metrics are employed, including accuracy, precision, recall, F1-score, and confusion matrix analysis. These metrics provide comprehensive insights into the capability of each model to correctly classify ransomware and benign traffic.Accuracy measures the proportion of correctly classified instances relative to the total number of samples. It is calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

where:

- TP (True Positive): ransomware correctly classified as ransomware
- TN (True Negative): benign traffic correctly classified as benign
- FP (False Positive): benign traffic incorrectly classified as ransomware
- FN (False Negative): ransomware incorrectly classified as benign

Precision measures the proportion of correctly predicted positive instances among all predicted positives:

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

Recall, also known as sensitivity, measures the proportion of correctly predicted positive instances among all actual positives:

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

The F1-score is the harmonic mean of precision and recall and is particularly useful for evaluating imbalanced datasets:

$$F1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{5}$$

In addition to numerical metrics, confusion matrices are used to visualize classification performance across all ransomware families. The confusion matrix provides detailed information regarding correctly and incorrectly classified samples, allowing deeper analysis of misclassification patterns between classes.The evaluation process compares the performance of LightGBM, Random Forest, and Decision Tree models under both baseline and optimized scenarios. The optimized models are expected to achieve better generalization performance due to the application of SMOTEENN and SelectKBest preprocessing techniques.

2.6. Research Results

The experiment utilized a Light Gradient Boosting Machine (LightGBM) classifier to identify ransomware by analyzing key features extracted from the dataset. This algorithm was selected due to its high performance, scalability, and suitability for large datasets with complex feature structures, which are commonly found in ransomware detection tasks. The implementation demonstrates a complete and optimized workflow, starting from preprocessing and model training to evaluation and model preservation.

The LightGBM implementation presents an optimized and production-ready classification pipeline tailored for ransomware detection. The combination of manual hyperparameter tuning, rigorous preprocessing, and comprehensive evaluation ensures that the model delivers high-quality predictions while remaining interpretable and suitable for deployment in real-world environments.

The model achieved an overall accuracy of 54.35%. In terms of macro-averaged and weighted metrics, the precision reached 54.13%, recall achieved 54.35%, and the F1-score obtained was 53.52%, as shown in Figure 3. These results indicate a moderate classification capability, but they also reveal inconsistencies in the model's effectiveness across different ransomware classes. While the model demonstrated promising precision for certain ransomware families, its ability to generalize across the full range of ransomware variants remained limited.

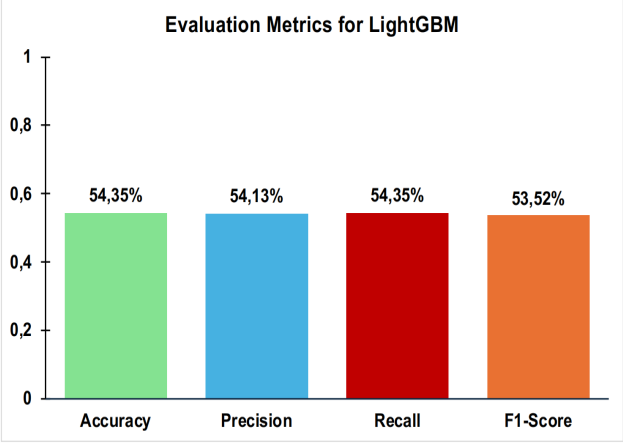

Figure 3: Evaluation Metrics for LightGBM Model

Based on the confusion matrix shown in Figure 4, the LightGBM model demonstrates strong performance on several dominant ransomware classes but struggles with minority classes. The *RANSOMWARE_CHARGER* class achieves the highest F1-score of 0.78, supported by a high recall value of 0.87. Similarly, *RANSOMWARE_SVPENG* shows strong recall performance of 0.70, although its relatively lower precision of 0.56 indicates a higher false positive rate.

In contrast, ransomware families such as *LOCKERPIN*, *SIMPLocker*, and *WANNALOCKER* exhibit relatively poor classification performance, with F1-scores below 0.45. This result suggests that the model experiences difficulty distinguishing these classes, likely due to overlapping behavioral features or insufficient training samples.

The *BENIGN* class achieves high precision of 0.84 but only moderate recall of 0.51, indicating that many benign samples are incorrectly classified as ransomware. Furthermore, the model shows a tendency to favor dominant classes such as *SVPENG*, which contains a large number of instances in the dataset, reflecting the presence of class imbalance bias.

Overall, although the LightGBM model demonstrates effectiveness in identifying several common ransomware families, it still faces challenges in generalizing to minority ransomware variants.

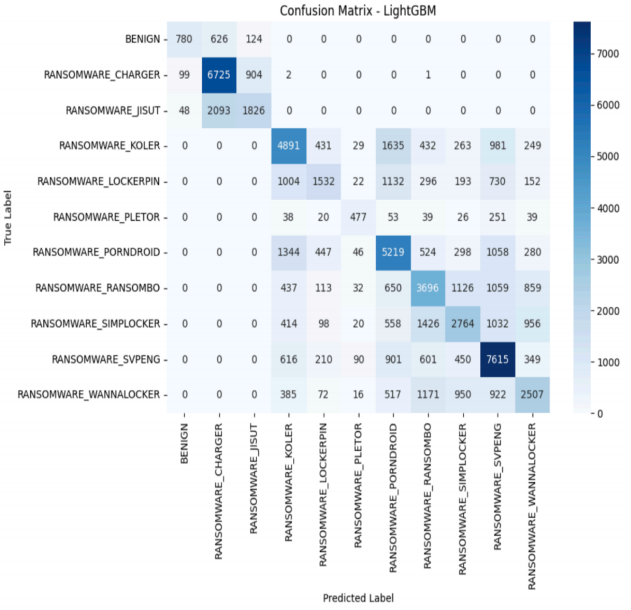

Figure 4: Confusion Matrix for LightGBM Model

The Random Forest classifier was implemented as one of the comparison models to evaluate its capability in multiclass ransomware detection. A `RandomForestClassifier` was instantiated with `n_estimators = 100`, indicating that the model consists of 100 decision trees within the ensemble. The model was trained using the `fit()` method on the training dataset `(X_train, y_train)`.

The Random Forest classifier achieved an overall accuracy of 47.12%. In terms of macro-averaged and weighted evaluation metrics, the precision reached 46.59%, recall achieved 47.12%, and the F1-score obtained was 46.44%, as illustrated in Figure 5.

These results indicate moderate classification performance in the multiclass ransomware detection scenario, with slightly better balance compared to the Decision Tree model. However, the achieved performance remains insufficient for practical real-world deployment involving complex and diverse ransomware behaviors.

Based on the confusion matrix shown in Figure 6, the Random Forest model exhibits moderate capability in classifying dominant ransomware families but continues to struggle with minority classes. The *RANSOMWARE_CHARGER* class demonstrates the highest performance, achieving an F1-score of 0.75, supported by a high recall value of 0.85 and a reasonable precision of 0.67. These results indicate that most instances belonging to this class are correctly detected.

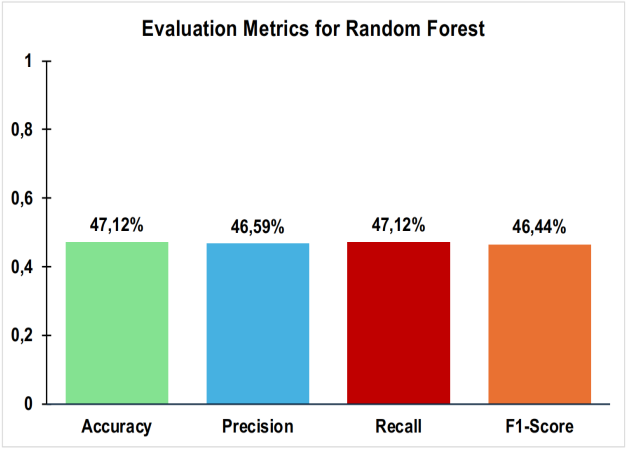

Figure 5: Evaluation Metrics for Random Forest Model

Another relatively well-classified ransomware family is *RANSOMWARE_SVPENG*, which achieves a recall of 0.59 and an F1-score of 0.56, although the model still experiences moderate misclassification into adjacent classes.

In contrast, ransomware families such as *RANSOMWARE_LOCKERPIN*, *SIMPLOCKER*, and *WANNALOCKER* exhibit weak classification performance, with F1-scores below 0.40. These results highlight the model's difficulty in learning representative patterns from underrepresented ransomware classes. This issue is particularly evident in the *RANSOMWARE_LOCKERPIN* class, where significant misclassification occurs across neighboring ransomware categories.

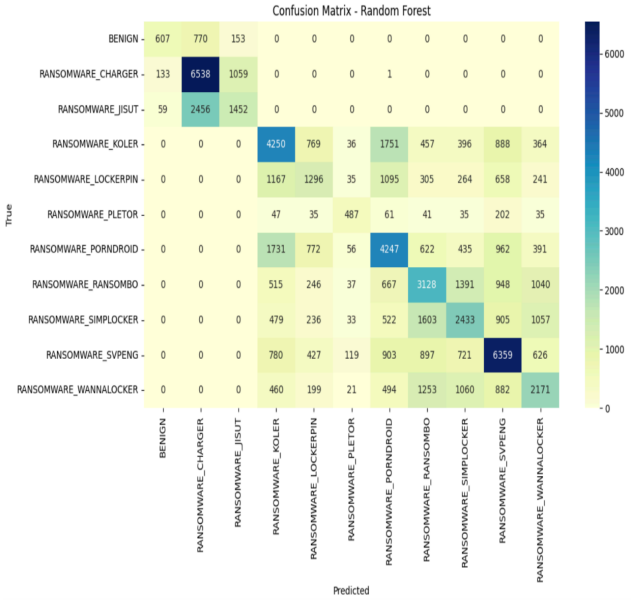

Figure 6: Confusion Matrix for Random Forest Model

The second comparison model employed in this study was the Decision Tree classifier. This classifier is particularly useful for interpretable machine learning, as it performs classification based on hierarchical decision rules derived from the input features.

A Decision Tree model was instantiated using `DecisionTreeClassifier(max_depth = 100)` with a fixed

`random_state` parameter to ensure reproducibility. The `fit()` method was then applied to train the model using the training dataset `(X_train, y_train)`. The `max_depth` parameter was used to limit the complexity of the tree and reduce the risk of overfitting, which is particularly important when dealing with high-dimensional network traffic data.

The Decision Tree classifier achieved an overall accuracy of 45.03%. In terms of macro-averaged and weighted evaluation metrics, the precision reached 45.10%, recall achieved 45.03%, and the F1-score obtained was 45.06%, as illustrated in Figure 7.

These results reflect a moderate but limited capability in handling the complexity of multiclass ransomware classification, particularly when distinguishing between closely related ransomware families.

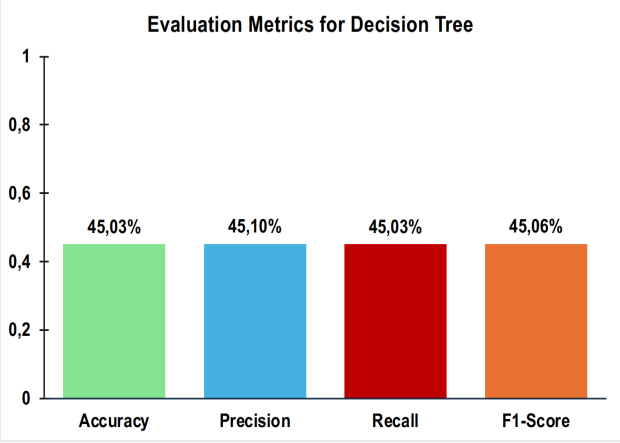

Figure 7: Evaluation Metrics for Decision Tree Model

Figure 8 presents the confusion matrix for the Decision Tree model, which demonstrates limited classification capability, particularly for minority ransomware families. Among all evaluated classes, *RANSOMWARE_CHARGER* achieves the best performance with an F1-score of 0.71, while the *BENIGN* and *RANSOMWARE_SVPENG* classes follow with F1-scores of 0.52 each, indicating moderate classification effectiveness.

In contrast, ransomware families such as *LOCKERPIN*, *SIMPLOCKER*, and *WANNALOCKER* suffer from severe misclassification, as reflected by their low F1-scores ranging from approximately 0.30 to 0.33. These classification errors are likely caused by overlapping feature characteristics and insufficient training samples for minority classes.

The confusion matrix also reveals several dominant misclassification patterns. For example, the *KOLER* class is frequently misclassified as *PLETOR*, *PORNDROID*, and *SVPENG*, while *RANSOMBO* is often incorrectly classified as *SIMPLOCKER* and *WANNALOCKER*.

Following the initial implementation of the LightGBM classifier, which yielded an accuracy of 54.35%, several refinements were undertaken to significantly enhance the model's performance. These improvements focused on structural design, generalization capability, and deployment feasibility. By introducing a modular architecture, employing effective data balancing techniques, integrating robust evaluation strategies, and incorporating model preservation procedures, the refined version of LightGBM emerged as a more dependable and production-ready solution for ransomware detection Kharraz et al. (2020); Woralert et al. (2024).

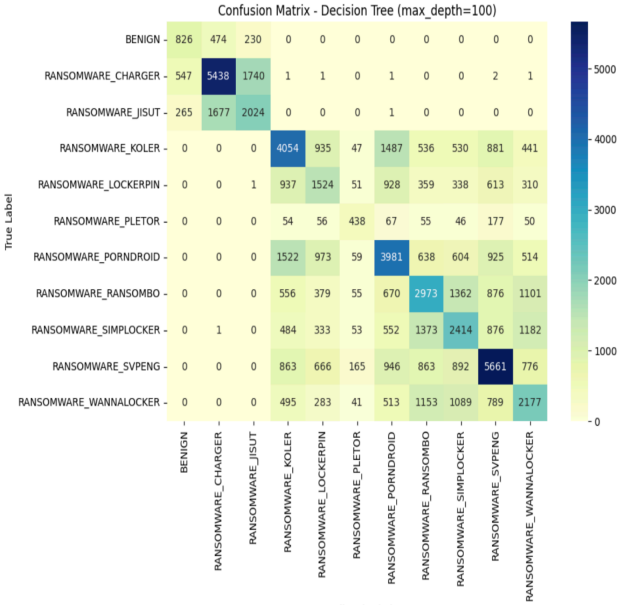

Figure 8: Confusion Matrix for Decision Tree Model

One of the key improvements involved addressing the significant class imbalance present in the original CIC-AndMal2017 dataset. Dominant classes such as *RANSOMWARE_SVPENG* and *BENIGN* contained tens of thousands of samples, while others such as *RANSOMWARE_PLETOR* had fewer than 5,000 instances. To mitigate this issue, the SMOTEENN technique was applied by combining oversampling of minority classes using SMOTE and noise reduction from majority classes using ENN.

This process reduced the dataset size from 349,855 to 193,172 instances while producing a more balanced label distribution. For example, the *RANSOMWARE_PLETOR* class increased to 42,006 samples, while the *BENIGN* class increased to 32,743 samples. This rebalancing process enabled the model to learn generalized patterns across all ransomware families, reducing classification bias and improving recall and F1-scores, particularly for previously underrepresented classes.

In addition to data balancing, the preprocessing pipeline was refined to improve data quality and learning efficiency. The baseline model initially utilized raw numerical features, which introduced scale-related bias during training. To address this issue, a `ColumnTransformer` pipeline was constructed using median imputation to handle missing values and `StandardScaler` for feature normalization, ensuring zero mean and unit variance across all features. This standardization process helped stabilize the training procedure and ensured fair contribution from all features.

Feature selection was another important enhancement. Although the original dataset included 84 network traffic-based features, many were identified as redundant or non-informative. Therefore, the `SelectKBest` method combined with ANOVA F-score analysis was employed to select the top 60 most relevant features Lv et al. (2024).

As illustrated in Figure 9, the top 20 most important features identified by the optimized LightGBM model included *Source Port*, *Flow IAT Min*, and *Flow Duration*. These features contributed significantly to the model's decision-making process, measured by total gain across all decision trees. Other high-impact attributes included *Init_Win_bytes_forward*, *Fwd IAT Min*, and *Flow Packets/s*, all of which played critical roles in capturing temporal and statistical variations between benign and malicious network traffic.

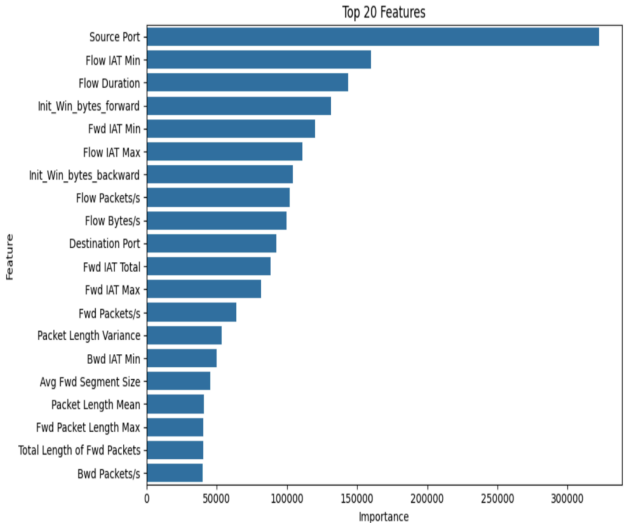

Figure 9: Top 20 Feature Importances of the Optimized LightGBM Model

Despite the reduction in input features, the optimized LightGBM model demonstrated remarkable performance improvements. Collectively, the integration of SMOTEENN, standardized preprocessing, selective feature reduction, and tuned hyperparameters led to a substantial increase in accuracy from 46.8% to 90.78%, while simultaneously reducing the feature space from 84 to 60 variables. These improvements not only enhanced classification precision but also increased computational efficiency and adaptability for real-world ransomware detection scenarios.

As shown in Figure 10, the optimized LightGBM model achieved an overall accuracy of 90.78%, with macro-averaged and weighted precision of 90.80%, recall of 90.78%, and an F1-score of 90.73%. These results demonstrate the model's strong capability to generalize across a wide range of ransomware families and benign traffic categories, even in the presence of significant data imbalance.

Figure 11 illustrates the confusion matrix of the optimized Light Gradient Boosting Machine (LightGBM) model using the SMOTEENN and SelectKBest techniques. The matrix demonstrates substantial improvements in classification performance across nearly all ransomware families, indicating that the integration of data balancing and feature selection significantly enhanced the model's generalization capability.

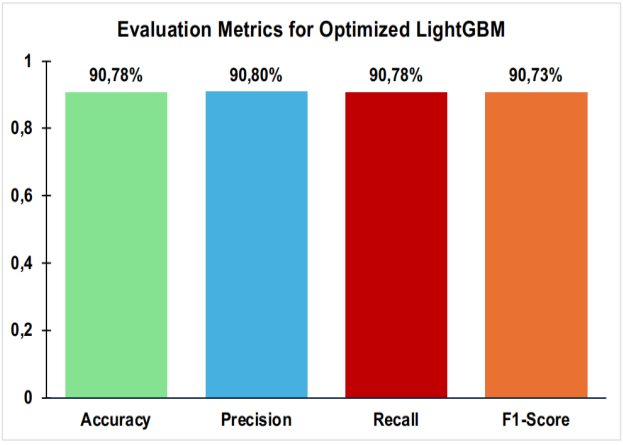

Figure 10: Evaluation Metrics for Optimized LightGBM Model

Figure 11 illustrates the confusion matrix for the Light Gradient Boosting Machine model optimized using SMOTEENN and SelectKBest, demonstrating robust classification performance across most ransomware families.

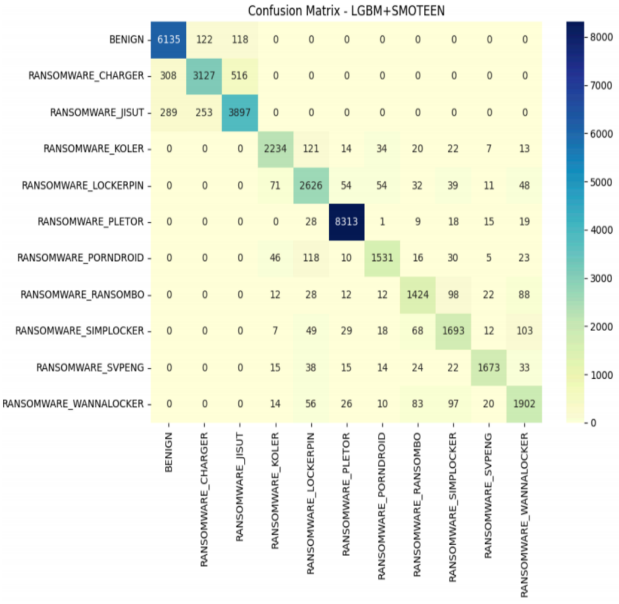

Figure 11: Confusion Matrix for Optimized LightGBM Model

For instance, the *BENIGN* class achieved a precision of 0.89 and a recall of 0.94, resulting in an F1-score of 0.91, which indicates the model's strong capability in distinguishing normal network traffic. The *RANSOMWARE_PLETOR* class achieved the highest performance with an F1-score of 0.99, reflecting near-perfect classification capability.

Other ransomware families such as *RANSOMWARE_LOCKERPIN*, *KOLER*, and *SVPENG* also demonstrated excellent performance, with F1-scores exceeding 0.90. Furthermore, ransomware variants that are traditionally more difficult to detect,

including *RANSOMWARE_WANNALOCKER* and *RANSOMWARE_RANSOMBO*, achieved strong F1-scores of 0.89 and 0.87, respectively.

The confusion matrix exhibits strong diagonal dominance, indicating highly accurate classification results. Minor misclassifications generally occurred among ransomware families with similar behavioral characteristics, such as *CHARGER* being misclassified as *JISUT*, and *SIMPLOCKER* overlapping with *WANNALOCKER* and *SVPENG*.

To ensure consistent preprocessing across all classification models, an additional training procedure was implemented using the SMOTEENN resampling strategy combined with `SelectKBest`-based feature selection, which retained only the 60 most informative features. This configuration was applied to both the Random Forest and Decision Tree classifiers to enable a reliable and fair comparative evaluation.

The performance evaluation of the Random Forest classifier trained using SMOTEENN and SelectKBest achieved a high overall accuracy of 89.65%, with a precision of 89.70%, recall of 89.65%, and a weighted F1-score of 89.55%, as illustrated in Figure 12.

These evaluation metrics indicate that the Random Forest model, when combined with balanced training data and selected informative features, is capable of delivering strong and stable performance in multiclass ransomware classification tasks.

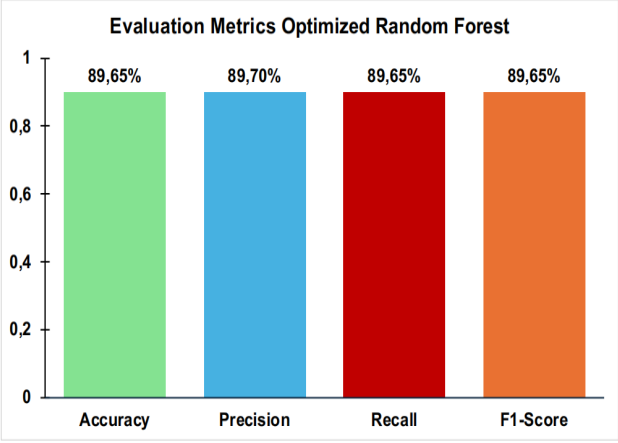

Figure 12: Evaluation Metrics for Optimized Random Forest Model

Figure 13 illustrates the confusion matrix of the Random Forest model optimized using SMOTEENN and SelectKBest, showing significant improvements in classification performance across nearly all ransomware families.

For example, the *BENIGN* class achieved a precision of 0.87 and a recall of 0.95, resulting in an F1-score of 0.91, confirming the model's ability to accurately distinguish non-malicious network behavior. Notably, the *RANSOMWARE_PLETOR* class achieved the highest F1-score of 0.99, indicating near-perfect precision and recall performance.

Other ransomware families, such as *RANSOMWARE_LOCKERPIN*, *RANSOMWARE_KOLER*, and *RANSOMWARE_SVPENG*, also demonstrated excellent performance, with F1-scores exceeding 0.90, indicating reliable detection capability for these ransomware variants.

Even ransomware families that are traditionally more challenging to classify, such as *RANSOMWARE_WANNALOCKER* and *RANSOMWARE_RANSOMBO*, achieved strong results with F1-scores of 0.87 and 0.85, respectively.

The confusion matrix further supports these findings by exhibiting strong diagonal dominance and minimal dispersion into off-diagonal cells. For instance, 6253 out of 6549 *BENIGN* samples and 8351 out of 8401 *PLETOR* samples were correctly classified, reflecting excellent predictive performance for both dominant and minority classes.

The remaining misclassifications generally occurred between ransomware families with semantically or behaviorally similar network traffic patterns. For example, several *RANSOMWARE_CHARGER* samples were incorrectly classified as *JISUT* or *BENIGN*, while*RANSOMWARE_SIMPLOCKER* demonstrated overlap with *WANNALOCKER* and *SVPENG*. These overlaps align with real-world similarities in ransomware network behavior.

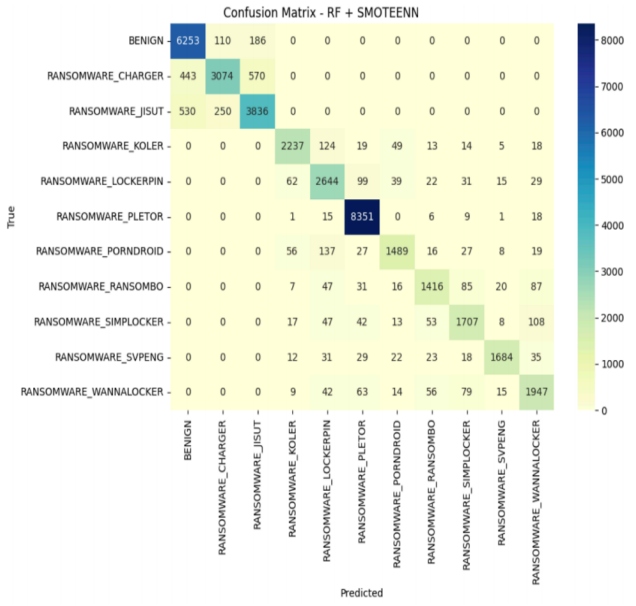

Figure 13: Confusion Matrix for Optimized Random Forest Model

In parallel, the following evaluation results were obtained from the optimized Decision Tree model trained using SMOTEENN and `SelectKBest` feature selection with the top 60 selected features. The performance of this configuration is illustrated in Figure 14. The optimized Decision Tree model achieved an overall accuracy of 82.12%, with a precision of 82.05%, recall of 82.12%, and a weighted F1-score of 82.07%, as presented in Figure 14. These results demonstrate strong and balanced performance in the multiclass classification of ransomware variants and benign network traffic.

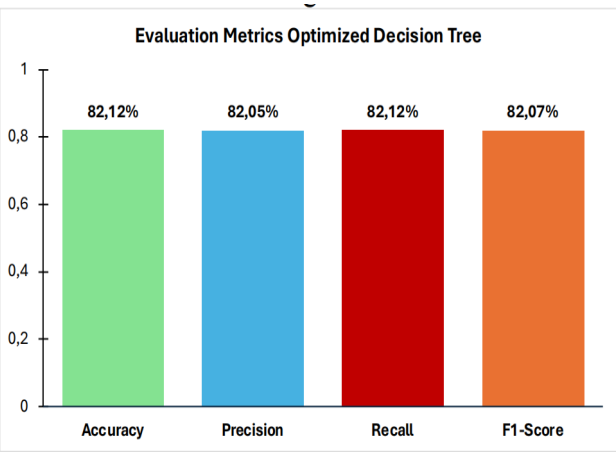

Figure 14: Evaluation Metrics for Optimized Decision Tree Model

Figure 15 illustrates the confusion matrix of the Decision Tree model optimized using SMOTEENN and SelectKBest, which demonstrates substantial improvements in classification performance across nearly all ransomware families.

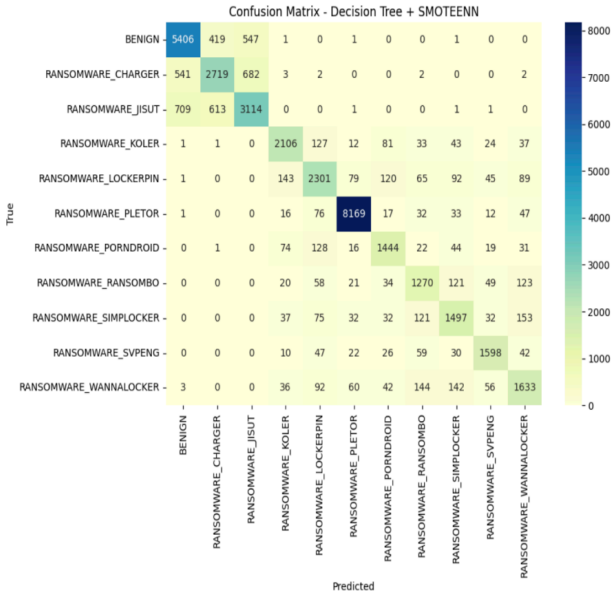

Figure 15: Confusion Matrix for Optimized Decision Tree Model

The strong diagonal dominance observed in the matrix indicates accurate predictions, particularly for ransomware classes that were previously difficult to classify due to class imbalance issues. The *RANSOMWARE_PLETOR* class achieved the highest performance with an F1-score of 0.97, followed by *RANSOMWARE_KOLER* with an F1-score of 0.86 and the *BENIGN* class with an F1-score of 0.83. These results reflect the model's improved capability to capture distinctive network traffic behaviors.

Furthermore, *RANSOMWARE_CHARGER* and *RANSOMWARE_JISUT* also demonstrated strong performance with F1-scores of 0.71 and 0.77, respectively, although minor confusion remained with adjacent classes such as *LOCKERPIN* and *BENIGN*.

Minority ransomware families, including *RANSOMWARE_PORNDROID* and *RANSOMWARE_RANSOMBO*, exhibited considerable improvement, achieving F1-scores of 0.79 and 0.74, respectively. These findings suggest that the SMOTEENN resampling strategy effectively addressed class imbalance by generating representative synthetic samples while simultaneously removing noisy instances.

This capability is particularly important in real-world ransomware detection scenarios, where rare ransomware variants must be accurately identified. Although several minor misclassifications still persisted, particularly among *SIMPLOCKER*, *SVPENG*, and *WANNALOCKER*, the overall error distribution remained relatively low.

Overall, the optimized Decision Tree model demonstrated balanced performance across both dominant and underrepresented ransomware classes, achieving a macro-average F1-score of 0.80 and a weighted F1-score of 0.82.

In the initial (non-optimized) condition, all 84 original features were utilized without applying any preprocessing techniques, class balancing strategies, or feature selection methods. As presented in Table 2, the LightGBM model achieved the best performance among the evaluated classifiers, with an overall accuracy of 54.35% and an F1-score of 53.52%.

Although these results demonstrate a moderate capability in detecting malicious ransomware behavior, the model still exhibited limitations in distinguishing minority ransomware classes. This limitation was likely caused by class imbalance and feature redundancy within the dataset.

The Random Forest and Decision Tree classifiers produced lower performance results. The Random Forest model achieved an accuracy of 47.12% with an F1-score of 46.44%, while the Decision Tree model obtained an accuracy of 45.03% and an F1-score of 45.06%.

These relatively suboptimal performances can be attributed to several challenges, including skewed class distribution, the presence of irrelevant or redundant features, and insufficient representation of rare ransomware samples within the dataset.

Table 2: Performance Comparison Before Optimization

Method	Accuracy	Precision	Recall	F1-Score	Features
LightGBM	54.35%	54.13%	54.35%	53.52%	79
Random Forest	47.12%	46.59%	47.12%	46.44%	80
Decision Tree	45.03%	45.10%	45.03%	45.06%	80

To address these limitations, an optimization pipeline was applied that included SMOTEENN (Synthetic Minority Oversampling Technique combined with Edited Nearest Neighbors) for class balancing and SelectKBest for feature selection. SMOTE was used to generate synthetic samples for underrepresented classes, thereby improving class balance during training. In parallel, ENN removed noisy or overlapping instances from the majority classes, resulting in a cleaner dataset. Feature selection via SelectKBest was employed

to reduce dimensionality by retaining the most informative features.

The impact of these optimization strategies is evident in Table 3, where all models demonstrated substantial improvements across all metrics. The optimized LightGBM model achieved 90.78% accuracy, 90.80% precision, 90.78% recall, and an F1-score of 90.73%, indicating a highly stable and balanced classification performance. This consistency suggests that the model is capable of accurately differentiating both ransomware families and benign traffic. Random Forest also improved significantly, achieving an accuracy of 89.65% and an F1-score of 89.55%. The Decision Tree model, which initially had the lowest performance, experienced a notable enhancement, reaching 82.12% accuracy and an F1-score of 82.07%.

Table 3: Performance Comparison After Optimization

Method	Accuracy	Precision	Recall	F1-Score	Features
Optimized LightGBM	90.78%	90.80%	90.78%	90.73%	60
Optimized Random Forest	89.65%	89.70%	89.65%	89.55%	60
Optimized Decision Tree	82.12%	82.05%	82.12%	82.07%	60

These findings highlight the critical role of preprocessing and optimization in building effective classification models for cybersecurity applications. Without these techniques, models tend to overfit dominant classes and fail to generalize well to underrepresented threats. In contrast, the optimized models, particularly LightGBM, achieved significant performance gains, with improved accuracy and generalization across all ransomware families.

Among the three algorithms, LightGBM consistently outperformed the others in both baseline and optimized scenarios. Its ability to maintain high precision and recall across multiple classes makes it a strong candidate for integration into real-time ransomware detection systems. The combination of SMOTEENN and SelectKBest not only enhanced the quality of the training dataset but also improved the interpretability and robustness of the models.

The experimental results confirm that optimization strategies are essential in addressing the challenges of imbalanced, high-dimensional datasets commonly found in cybersecurity. The superior performance of the optimized LightGBM model demonstrates its potential for operational deployment and sets a foundation for future research involving ensemble learning, advanced feature engineering, and real-time detection frameworks.

3. Conclusion and Recommendation

3.1. Conclusion

This research aimed to develop a robust machine learning framework for multiclass ransomware detection using the Light Gradient Boosting Machine (LightGBM) algorithm. The study also benchmarked its performance against two widely used classifiers: Random Forest and Decision Tree. The primary objective was achieved by training and evaluating these models using the CIC-AndMal2017 dataset, which contains diverse network traffic data representing both ransomware and benign behavior.

In the baseline scenario, minimal preprocessing was applied, including low-cardinality categorical encoding, removal of irrelevant metadata columns such as IP addresses and timestamps, and handling of missing values. All original 84 features were retained without applying feature selection or resampling. Under these conditions, the LightGBM model achieved an initial accuracy of 54.35% using 79 numeric features. Although this performance indicates a preliminary ability to detect ransomware activity, the model faced challenges in generalizing to minority ransomware classes due to class imbalance and the presence of redundant or less informative features.

To improve classification performance and model generalization, two key optimization techniques were applied: SMOTEENN for class balancing and SelectKBest for feature selection. This pipeline reduced the number of features from 84 to 60 while producing a more balanced and cleaner training dataset. As a result, the optimized LightGBM model showed a substantial improvement, attaining an accuracy of 90.78%, precision of 90.80%, recall of 90.78%, and a weighted F1-score of 90.73%. These results affirm the model's effectiveness in handling complex, high-dimensional, and imbalanced network traffic data commonly associated with ransomware attacks.

For a comprehensive comparison, the performance of LightGBM was benchmarked against Random Forest and Decision Tree under both non-optimized and optimized conditions. The evaluation results consistently demonstrated that LightGBM outperformed both alternative models across all key performance metrics. After optimization, Random Forest achieved an accuracy of 89.65% and an F1-score of 89.55%, while Decision Tree attained 82.12% accuracy and an F1-score of 82.07%. These findings confirm that the optimized LightGBM model delivers superior performance in multiclass ransomware detection tasks and is well suited for operational deployment in cybersecurity systems.

Furthermore, this research highlights the pivotal role of data preprocessing, particularly feature selection and class balancing, in enhancing model accuracy, stability, and interpretability. Without these optimization steps, model performance remained moderate and biased toward dominant classes. By contrast, the optimized models, especially LightGBM, demonstrated significantly improved capability to distinguish both rare ransomware variants and benign traffic with high precision.

3.2. Recommendation

Despite the encouraging results obtained from the optimized LightGBM model, further improvements are necessary to meet the rigorous demands of real-world deployment in critical domains such as finance, healthcare, and government infrastructure. Future research should consider expanding the dataset to include a wider and more up-to-date range of ransomware variants. This could be achieved by incorporating samples from alternative sources such as Drebin, AndroZoo, or actual incident response logs. Such diversification is expected to enhance the model's ability to generalize to

emerging and previously unseen threats Ajayi and Chigbu (2025).

In addition, incorporating dynamic behavioral features such as system call traces, file system interactions, and sandbox execution logs could complement the current static network-based features. A multimodal feature approach may provide a richer understanding of ransomware behavior and lead to improved detection accuracy.

Future studies are also encouraged to explore hybrid or ensemble modeling strategies by combining LightGBM with other algorithms such as Convolutional Neural Networks (CNNs), Support Vector Machines (SVMs), or Transformer-based models. Ensemble techniques such as stacking, majority voting, or boosting could reduce false positives and enhance robustness, particularly in adversarial or evolving threat environments Kauser and Showkath Ali (2025).

Finally, integrating the proposed model into a broader layered defense system alongside traditional detection techniques such as signature-based methods, heuristic analysis, and anomaly detection would improve its practicality for real-time security applications. A layered security approach not only broadens detection coverage but also provides greater adaptability and resilience in dynamic and complex cyber threat landscapes.

References

Ajayi, O. and Chigbu, D. I. (2025). Adversarial machine learning in cybersecurity: Challenges and countermeasures. *Electronics*, 14(3):590.

Albin Ahmed, A. and Shaikh, K. (2023). Android ransomware detection using supervised machine learning techniques based on traffic analysis. *Bulletin of Electrical Engineering and Informatics*, 12(2):1049–1056.

AlHashmi, A., Darem, A., and Abawajy, J. H. (2024). Ransomware early detection: A survey. *Future Generation Computer Systems*, 161:103–122.

Ali, H., Pal, A., et al. (2024). Empowering ransomware detection with machine learning and ensemble methods: A comprehensive feature analysis. *IEEE Access*, 12:194879–194892.

Aljabri, M., Altamimi, S. S., et al. (2024). Ransomware detection based on machine learning using memory features. *Egyptian Informatics Journal*, 25:100445.

Alkasassbeh, M. and Abbadi, M. A. (2020). Lightgbm algorithm for malware detection. *International Journal of Intelligent Information and Database Systems*, 16(2):1–14.

Avhankar, P. B. and Khandare, S. S. (2025). A comprehensive survey on the taxonomy of malware detection techniques. *Measurement: Sensors*, 37:101355.

Cen, M., Jiang, F., and Doss, R. (2024). Zeroran: Zero-day ransomware detection with feature engineering adaptation. In *Proceedings of the 2024 International Conference on Computing, Networking and Communications (ICNC)*, pages 375–380.

Chen, J., Wang, C., Zhao, Z., et al. (2018). Uncovering the face of android ransomware: Characterization and real-time detection. In *Proceedings of IEEE INFOCOM 2018*, pages 386–394.

Coveware (2021). Ransomware payments decline as fewer companies pay. https://www.coveware.com/blog/2021/2/1/ransomware-payments-decline-as-fewer-companies-pay.

Gao, C., Weng, G., Jia, L., and Du, J. (2022). Malware detection using lightgbm with a custom loss function. *IEEE Access*, 10:84656–84669.

Hagerty, D. and Nygard, K. (2024). Ransomware trends and mitigation strategies. *Journal of Cybersecurity Education, Research and Practice*, 2024(2):5.

Hussain, F., Abbas, S. G., et al. (2025). Ransomware detection and classification using machine learning. *Computers, Materials & Continua*, 82(2):3175–3201.

Imran, M., Afzal, M. T., and Qadir, M. A. (2024). Evaluating machine learning classifiers for android malware detection. *Intelligent Decision Technologies*, 18(2):1133–1153.

Kalhana, R. S., Gadekalu, T. R., et al. (2024). Prediction and mitigation of ransomware attacks using machine learning. *Peer-to-Peer Networking and Applications*, 17:3045–3060.

Kara, I. (2023). Fileless malware threats: Recent advances, analysis approach through memory forensics and research challenges. *Expert Systems with Applications*, 214:119133.

Kauser, S. K. and Showkath Ali, M. (2025). Hybrid ensemble-based machine learning approach for ransomware detection. *International Journal of Computational Intelligence Systems*, 18:49.

Kharraz, A., Robertson, W., and Kirda, E. (2020). Redemption: Real-time protection against ransomware at end-hosts. *ACM Transactions on Privacy and Security*, 23(3):1–30.

Kirubavathi, G. and Deepa, N. (2024). Behavioural analysis and classification of android malware using machine learning. *Multimedia Tools and Applications*, 83:73291–73314.

Lv, Z., Fang, L., et al. (2024). Ctimd: Cyber threat intelligence enhanced android malware detection using feature selection. *IEEE Transactions on Information Forensics and Security*, 19:6797–6811.

Madanayaka, M. et al. (2023). A proactive approach against android ransomware: Frameworks, detection, and analysis. *IEEE Access*, 11:81978–82004.

Momposhi, L., Isafiade, O., and Bagula, A. (2025). A comparative study of machine learning algorithms for android ransomware detection. *Computers & Security*, 147:104094.

Nguyen, Q. U. and Lee, S. (2021). Lightgbm-based ransomware detection using api call sequences. *Applied Sciences*, 11(13):6068.

Sangher, L. S., Bhatia, A., and Singh, K. (2024). Signature-based malware detection: An imminent concern. *International Journal of Network Security & Its Applications*, 16(5):97–110.

Stritch, T. and Allyn, J. (2021). The conti ransomware gang: An overview. https://www.cisa.gov/news-events/cybersecurity-advisories/aa21-265a.

Woralert, C., Liu, C., and Zuo, Z. (2024). Towards robust ransomware detection: A deep learning and explainability approach. *SN Computer Science*, 5:365.

Zahoora, U., Khan, A., et al. (2022). Ransomware analysis using cyber kill chain-based taxonomy and machine learning. *Computers & Security*, 115:102618.

Zhou, Q. et al. (2024). A novel android malware detection method using lightgbm. *Neural Computing and Applications*, 36:12301–12316.