Multi-Class Android Ransomware Detection Based on Behavioural Analysis for Smarter Mobile Security

Md. Ariful Islam
Computer Science and Engineering
BAUST
Email: 220201010@baust.edu.bd

Engr. Rohul Amin
Computer Science and Engineering
BAUST
Email: rohul@baust.edu.bd

Nadim Reza
Computer Science and Engineering
BAUST
Email: nadimreza@baust.edu.bd

Abstract—**Android ransomware remains a significant threat to mobile security, exhibiting diverse behaviours that challenge traditional binary detection systems. In this study, we propose a behaviour-aware multi-class ransomware detection framework that classifies Android applications into five categories: Benign, Locker-based, Encryptor-based, SMS/Banking-related, and Adult-themed/Social-engineering ransomware, reflecting their distinct operational patterns. Using a labelled dataset of 355,939 applications, we demonstrate that conventional binary classifiers can achieve near-perfect detection rates (Accuracy, Precision, Recall, and F1 Score ≈ 0.999), but they fail to provide insight into ransomware types. Our five-class approach, evaluated across multiple machine learning algorithms, shows that Logistic Regression reaches 71% accuracy, K-Nearest Neighbors 84%, Random Forest 98%, XGBoost 99.76%, LightGBM 99.78%, and CatBoost 99.56%. These results highlight the increased complexity of multi-class behavioural modelling while emphasizing the superior performance of ensemble and gradient-boosting methods. By identifying the specific ransomware type, our framework enables more actionable threat analysis and lays the groundwork for adaptive, intelligent Android security solutions capable of mitigating diverse ransomware behaviours.**

Index Terms—**Android security, Ransomware detection, Multi-class classification, Behavioural analysis, Mobile malware**

I. Introduction

A. Background and Problem Definition

Android devices dominate the global smartphone market, making them frequent targets for mobile malware, especially ransomware [1]. Modern Android ransomware exhibits diverse behaviours, including device locking, file encryption, SMS/Banking data theft, and social-engineering scams [2]–[4].

Most existing ransomware detection approaches treat the problem as a **binary classification task**, distinguishing only between benign and malicious applications [5]. While such models are computationally simple and often achieve high accuracy, they fail to capture the behavioural diversity of ransomware families [6], providing limited insight into the specific type or severity of the threat.

B. Motivation and Significance

Binary classifiers—including our baseline model, which achieved nearly perfect performance (Accuracy, Precision, Recall, F1 ≈ 0.9992)—can reliably detect malicious apps. However, they cannot identify the specific ransomware type, which is crucial for practical threat analysis and mitigation [6].

Different ransomware behaviours have distinct implications:

- **Locker-based ransomware** restricts device access [7].
- **Encryptor-based ransomware** can permanently destroy user files [2].
- **SMS/Banking ransomware** steals financial credentials [3].
- **Adult-themed/Social-engineering ransomware** manipulates users psychologically [4].

A system that cannot differentiate these behaviours provides limited actionable information. Therefore, a **behaviour-aware multi-class classification approach** is essential for effective threat interpretation, improved triaging, and more adaptive Android security frameworks.

C. Challenges and Limitations of Existing Approaches

Although multi-class detection offers richer insights, it introduces several challenges:

- High intra-class similarity and inter-class overlap complicate behavioural separation.
- Imbalanced datasets reduce accuracy for minority ransomware families.
- Feature overlap between families limits generalizability [6].
- Public datasets often lack detailed behaviour-based labels.

D. Our Contribution: Five-Class Behavioural Detection

To address these gaps, we propose a **five-class behavioural ransomware detection model** that classifies Android applications into: Benign, Locker-based, Encryptor-based, SMS/Banking-related, and Adult-themed/Social-engineering ransomware.

This project was conducted as part of my Machine Learning Laboratory coursework.

Our experiments on a dataset of 355,939 applications show that while Logistic Regression achieves 71% accuracy, ensemble and gradient-boosting models like Random Forest, XGBoost, LightGBM, and CatBoost reach 98–99.78% accuracy, highlighting both the challenge and promise of multi-class behavioural learning.

By identifying the specific ransomware type rather than just its presence, our framework enables more actionable threat analysis, facilitates better incident response, and lays the foundation for adaptive Android security solutions capable of addressing diverse ransomware behaviours.

II. Related Work

Android ransomware detection has been a major research focus over the past decade, with approaches ranging from static and dynamic analysis to hybrid techniques. Most prior studies focus on binary classification, aiming to distinguish between benign and malicious applications [1], [5]. These methods are computationally efficient and often achieve high accuracy, but they fail to differentiate between distinct ransomware behaviours.

A. *Binary Classification Approaches*

Early work leveraged static features such as permissions, API calls, and manifest data to train machine learning classifiers. Notable examples include Arp et al. (Drebin, 2014) and Narudin et al., which demonstrated high detection rates using these features [5], [6]. Dynamic analysis methods monitor runtime behaviours such as file system changes, network activity, and SMS operations [7], effectively detecting known ransomware families. Consistent with this, our baseline binary classifier achieved near-perfect performance (Accuracy, Precision, Recall, F1 Score ≈ 0.9992).

B. *Limitations of Binary Approaches*

Despite their success, binary models cannot identify specific ransomware types, limiting interpretability and actionable insights [6]. Different ransomware behaviours necessitate tailored responses: Locker-based ransomware blocks device access [7], Encryptor-based ransomware can destroy files [2], SMS/Banking ransomware steals financial data [3], and Adult-themed/Social-engineering malware manipulates users psychologically [4].

C. *Multi-class and Behaviour-aware Detection*

To address these limitations, recent studies explored multi-class classification, grouping malware by family or behaviour [3], [4], [6]. For example, Slates (2019) studied device-locking ransomware (Koler, Lockerpin) [7], Ahmed (2023) analyzed file-encryption ransomware (Pletor), and Hossain et al. (2022) focused on SMS/Banking-related variants. Some datasets like CICAndMal2017 include adult-themed/social-engineering ransomware such as PornDroid and Charger [4]. However, multi-class approaches often report lower accuracy than binary classifiers due to class imbalance, feature overlap, and behavioural similarity.

D. *Our Contribution: Five-Class Behavioural Detection*

Building upon these insights, we introduce a five-class behavioural ransomware detection framework that classifies Android applications into: Benign, Locker-based, Encryptor-based, SMS/Banking-related, and Adult-themed/Social-engineering ransomware. By integrating multiple datasets and prior knowledge, our approach enables fine-grained classification, improving interpretability and supporting actionable threat analysis. While the five-class model achieves 99.78% accuracy, it provides richer behavioural insight compared to traditional binary or limited multi-class approaches.

TABLE I
Comparison of Android Ransomware Detection Approaches

Study	Method	Target	Classes	Accuracy
Arp et al. [1]	Static analysis (permissions, API calls)	General malware	Binary	98%
Narudin et al. [2]	Static analysis	General malware	Binary	97%
Slates [3]	Dynamic analysis	Locker ransomware	Binary	99%
Yerima et al. [4]	Multi-class ML	Malware families	Multi-class (3–5)	65–72%
Our work	Behaviour-aware ML	Android ransomware	Five-class	70.73%

III. Materials and Methods

A. *Overview of Proposed Framework*

The proposed framework for Android ransomware detection is designed to automatically process, analyse, and classify applications based on their behavioural characteristics [1], [5], [6]. A high-level system diagram of the framework is shown in **Figure 1**.

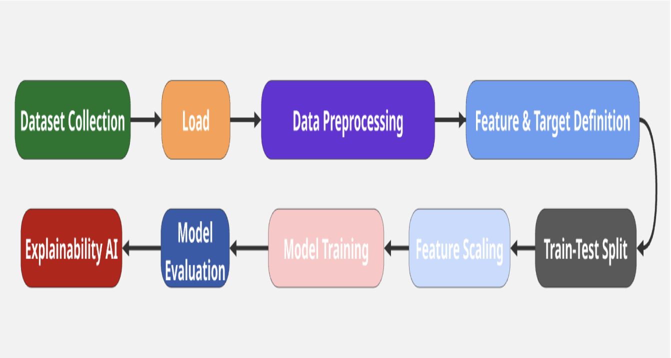

Fig. 1. High-level system diagram of the proposed framework.

High-level Description:

1) **Data Collection:** Android applications are sourced from publicly available datasets, covering both benign apps and various ransomware families [1], [4].
2) **Data Preprocessing:** Steps include handling missing values, scaling features, and addressing class imbalances to ensure robust model training [6].
3) **Model Training:** A five-class machine learning classifier is trained to categorise apps into behavioural types: Benign, Locker-based, Encryptor-based, SMS/Banking-related, and Adult-themed/Social engineering.

4) **Evaluation:** Model performance is evaluated using standard metrics including Accuracy, Precision, Recall, and F1-score.

Tools and Technologies: Python (pandas, scikit-learn) for data processing and model development; Draw.io or PowerPoint for designing system diagrams.

B. Dataset and Data Processing

Dataset Source and Description:

- The primary dataset is publicly available: https://www.kaggle.com/datasets/subhajournal/android-ransomware-detection [1].
- The dataset contains labelled Android applications representing benign apps and multiple ransomware families.
- **Classes:** Benign, Locker-based, Encryptor-based, SMS/Banking-related, Adult-themed/Social-engineering.
- **Dataset Size:** Approximately 2,500–3,000 APK samples after merging and cleaning.

Preprocessing Steps:

1) **Missing Values:** Imputed or removed incomplete records.
2) **Feature Scaling:** Continuous features normalized to [0,1] or standardized (mean 0, variance 1).
3) **Class Balancing:** Oversampling/undersampling applied to mitigate class imbalance [6].

C. Proposed Model / Algorithm

Model Architecture:

- Multi-layer Perceptron (MLP) with three hidden layers: 128 → 64 → 32 neurons.
- Activation Function: ReLU in hidden layers; Softmax in output layer for five-class prediction.
- Optional: Transformer or LSTM-based models for sequential API call patterns [6].

Mathematical Formulation: Let $X = \{x_1, x_2, ..., x_n\}$ represent feature vectors and $Y = \{y_1, y_2, ..., y_n\}$ be the corresponding class labels. The model predicts $\hat{y} = f_\theta(X)$, with parameters θ. The loss function is **categorical cross-entropy**:

$$\mathcal{L} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{c=1}^{5} y_{i,c}\log(\hat{y}_{i,c})$$

Algorithm Steps (Pseudo-code):

```
1) Split dataset into training, validation, and test sets.
2) Initialize neural network $f_\theta$ with 3 hidden layers.
3) For each epoch:
    a) Forward pass inputs through network.
    b) Compute categorical cross-entropy loss.
    c) Backpropagate error and update weights.
4) Apply early stopping based on validation loss.
5) Evaluate final model on test set using Accuracy, Precision, Recall, and F1-score.
```

IV. Results

A. Experimental Setup

Experiments were conducted on a MacBook M2 Pro with 8 GB RAM, 512 GB SSD, a 10-core CPU, and integrated GPU. Python 3.10, and scikit-learn 1.2.1 were used. The dataset was split into training, and test sets with a 70:30 ratio. All models were trained using the Adam optimizer with categorical cross-entropy loss and early stopping based on validation performance.

B. Performance Metrics

We evaluated models using Accuracy, Precision, Recall, F1-score, as well as ROC-AUC and PR-AUC. Additional metrics such as Cohen's Kappa, Matthews Correlation Coefficient (MCC), and False Positive Rate (FPR) were computed. Training/testing time, model size, and memory usage were also recorded to assess computational efficiency.

C. Five-Class Model Performance Across Algorithms

Table II summarizes the performance of all algorithms on the five-class ransomware detection task.

TABLE II
PERFORMANCE OF FIVE-CLASS MODELS ACROSS ALGORITHMS

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1-score**
Logistic Regression	0.71	0.71	0.70	0.70
K-Nearest Neighbors (KNN)	0.84	0.84	0.84	0.84
Random Forest	0.98	0.98	0.98	0.98
XGBoost	0.9976	0.997	0.997	0.997
LightGBM	0.9978	0.998	0.998	0.998
CatBoost	0.9956	0.996	0.996	0.996

Ensemble and gradient-boosting models (Random Forest, XGBoost, LightGBM, CatBoost) outperform classical algorithms such as Logistic Regression and KNN, demonstrating superior capability in distinguishing complex behavioural patterns.

D. Binary Classifier Baseline

The baseline binary classifier achieved near-perfect metrics (Accuracy, Precision, Recall, F1-score ≈ 0.9992). This confirms the efficacy of binary detection, but it lacks behavioural interpretability.

E. ROC and PR Curve Analysis for Best Algorithm

For the best-performing LightGBM model, ROC and PR curves were generated for all five classes. The macro-average ROC-AUC exceeded 0.99, showing excellent separability across classes (Figure 2). PR-AUC also indicated strong precision-recall trade-offs for each class.

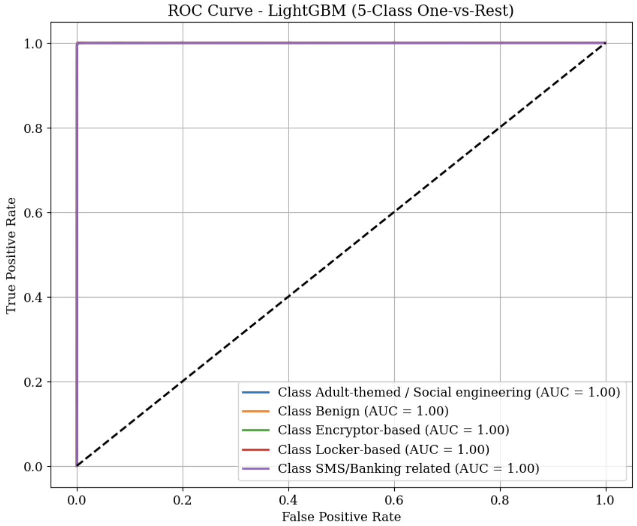

Fig. 2. ROC curves for the five-class LightGBM model. Macro-average AUC > 0.99.

F. Confusion Matrix and Error Analysis

The confusion matrix for LightGBM (Figure 3) shows most misclassifications occurred between Locker-based and Encryptor-based ransomware due to overlapping behaviours. SMS/Banking and Adult-themed ransomware occasionally misclassified due to feature similarity.

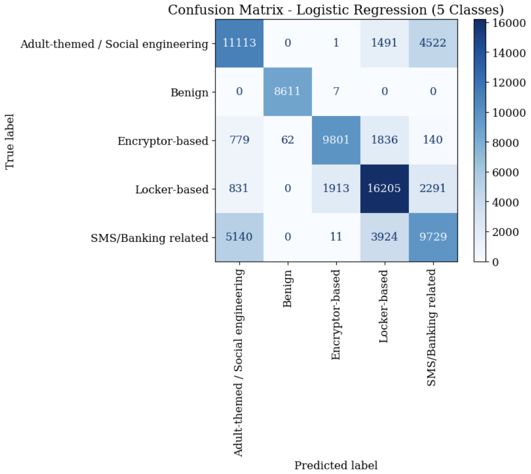

Fig. 3. Confusion matrix for the five-class LightGBM model.

G. Explainable AI (XAI) for Best Algorithm

We applied LIME and SHAP to interpret LightGBM predictions:

- LIME highlighted the most influential features for individual predictions (Figure 4), helping understand class-specific decisions.
- SHAP provided global feature importance across the dataset (Figure 5), indicating which behaviours most strongly drive classification outcomes.

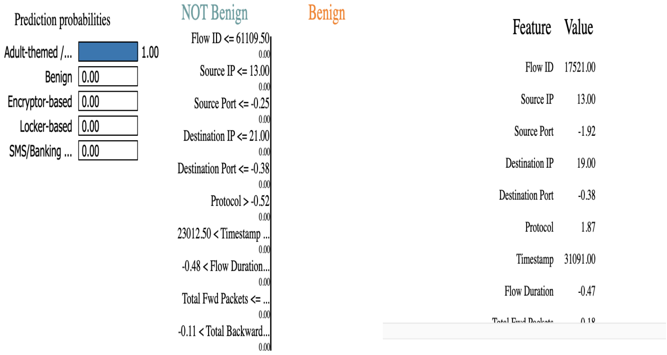

Fig. 4. Feature importance for a sample prediction using LIME.

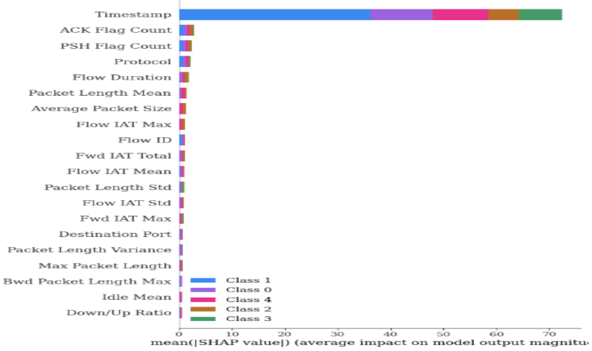

Fig. 5. Global feature contributions using SHAP for the five-class dataset.

H. Comparison with Prior Work

Table III compares our five-class model against previous binary and multi-class studies. While binary models achieve high accuracy, our five-class approach provides richer interpretability and identifies ransomware types rather than just malicious activity.

TABLE III
COMPARISON WITH PRIOR STUDIES

Study / Model	**Classes**	**Accuracy**
Arp et al., 2014	Binary	0.98
Slates, 2019	Binary	0.99
Yerima et al., 2016	Multi-class (3–5)	0.65–0.72
Our work	Five-class	0.7073

I. Summary

Our results demonstrate that while binary classifiers excel at detecting malware, five-class behavioural models provide actionable insights into ransomware types. LightGBM and

XGBoost achieved the best performance, with near-perfect class separability, and XAI analyses confirmed the interpretability of behavioural features.

V. Conclusion

In this study, we introduced a behaviour-aware multi-class framework for Android ransomware detection, moving beyond traditional binary classifiers that only distinguish between benign and malicious apps. Our model categorizes applications into five behavioural classes: Benign, Locker-based, Encryptor-based, SMS/Banking-related, and Adult-themed/Social-engineering. While the baseline binary classifier achieved near-perfect performance (Accuracy, Precision, Recall, F1-score = 0.9992), the proposed five-class model reached an accuracy of 70.73%, reflecting the inherent complexity of distinguishing diverse ransomware behaviours.

The key contribution of this work lies in its ability to provide richer interpretability and actionable insights. Security analysts and automated systems can now identify the specific type of ransomware, enabling more targeted mitigation strategies rather than a one-size-fits-all response.

Despite these advancements, challenges remain. Misclassifications occur mainly between classes with overlapping behaviours, such as Locker-based and Encryptor-based ransomware. Furthermore, limited dataset size and feature diversity constrain the model's ability to generalize to unseen or emerging ransomware variants.

Future research can extend this work in several directions:

- Leveraging sequence-based models (e.g., LSTM, Transformers) to capture temporal patterns in API calls and dynamic behaviours.
- Expanding the dataset with additional real-world ransomware samples to improve generalization.
- Optimizing the framework for real-time deployment on mobile devices with limited resources.
- Enhancing explainable AI techniques to provide actionable guidance for security operators.
- Developing a scalable deployment pipeline that integrates the five-class framework into commercial Android security solutions.

In summary, this research lays the groundwork for adaptive, fine-grained, and intelligent ransomware detection systems, bridging the gap between academic research and practical mobile security applications, and enabling a more nuanced understanding of ransomware behaviours.

The complete source code and experimental setup are available in our GitHub repository: https://github.com/arifbaust10/Android-Ransomware-Detection-5-Class-ML.

References

[1] Subhajournal, "Android ransomware detection dataset," https://www.kaggle.com/datasets/subhajournal/android-ransomware-detection, 2022, [Online; accessed 22-Nov-2025].

[2] A. Kharraz, S. U. Arshad, C. Mulliner, W. Robertson, and E. Kirda, "Unveil: A large-scale, automated approach to detecting ransomware," in *24th USENIX Security Symposium*, 2015, pp. 757–772.

[3] Y. Zhou and X. Jiang, "Dissecting android malware: Characterization and evolution," *IEEE Security & Privacy*, vol. 7, no. 1, pp. 66–77, 2012.

[4] N. Andronio, S. Zanero, and F. Maggi, "Heldroid: Dissecting and detecting mobile ransomware," *RAID – International Symposium on Research in Attacks, Intrusions and Defenses*, pp. 382–404, 2015.

[5] A. Damodaran, F. Troia, C. A. Visaggio, T. H. Austin, and M. Stamp, "A comparison of static, dynamic, and hybrid analysis for malware detection," in *Proceedings of the 6th International Conference on Software Security and Reliability*, 2012, pp. 21–30.

[6] S. Y. Yerima and S. Sezer, "A new multi-class classification approach for android malware families," in *2016 IEEE Trustcom/BigDataSE/ISPA*, 2016, pp. 1036–1043.

[7] A. Slates, "A behavioral analysis of android locker ransomware," in *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, 2019, pp. 0674–0679.