AI-Augmented Real-Time Detection System for Android Ransomware

Project Report

Supervisor

Dr. Syed Adeel Ali Shah

Aamir Khan

DEPARTMENT OF COMPUTER SCIENCE & IT
UNIVERSITYOFENGINEERING&TECHNOLOGY
PESHAWAR

March, 2026

Table Of Content

1) Introduction

Android is the most widely used smartphone operating system, and its open-source nature makes it a common target for cyberattacks. One of the most harmful threats is Android ransomware, which locks the device or encrypts user files and then demands money (usually through cryptocurrency) to restore access. Such attacks can cause serious data loss, financial damage, and loss of user trust.

Many traditional antivirus systems rely on signatures, so they often fail against new ransomware variants that use obfuscation and evasion techniques. Also, ransomware acts very fast, so offline or batch-based detection becomes ineffective because it detects the threat after damage has already started. Therefore, this project proposes a real-time, AI-assisted detection system that can classify Android app behaviour as benign or ransomware with high accuracy and low delay. The system uses an ensemble machine learning model (XGBoost) and a streaming pipeline to support real-time decision-making.

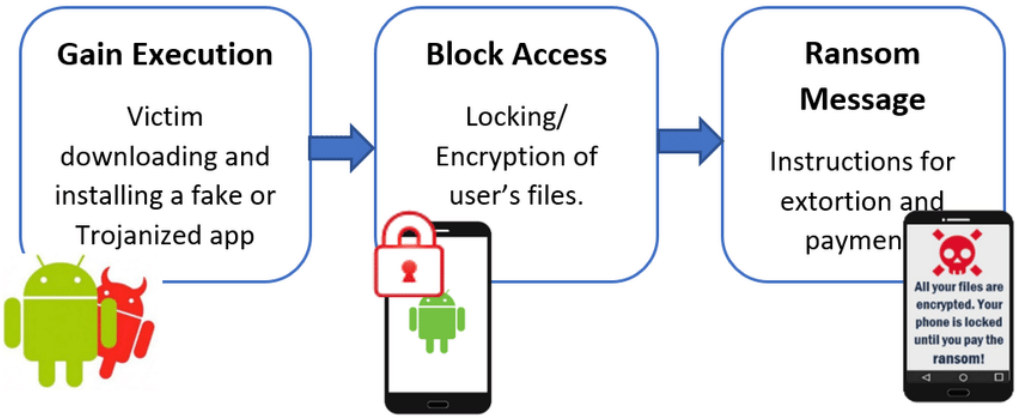


Figure 1 : Lifecycle of Android ransomware

2) Aim and Objectives

Derive and test a real time ensemble based machine learning system to detect Android ransomware.

Objectives

1) To review existing ensemble-based ML techniques for Android ransomware

detection and analyse their limitations in real-time contexts.

2) To propose a real-time supervised ensemble-based framework for Android ransomware detection.
3) To implement the proposed framework using the CIC (Canadian Institute for Cybersecurity) MalDroid 2020 dataset and streaming technologies (e.g., Apache Kafka, Docker).
4) To evaluate the framework in terms of accuracy, latency, and throughput.

To benchmark the proposed framework against existing offline detection mechanisms.

3) Problem Statement

Although many machine learning models show high accuracy in detecting Android ransomware, most existing solutions are tested in offline environments and use batch processing. This is a major limitation because ransomware can lock the device or encrypt files within seconds, and delayed detection may not prevent the attack.

Therefore, the main problem addressed in this project is:

How can a real-time supervised machine learning framework be designed and evaluated to detect Android ransomware accurately while maintaining low latency in a streaming environment?

4) Methodology

This study adopted a quantitative experimental research design based on supervised machine learning techniques. The CICMalDroid 2020 dataset was used for model development and evaluation, as it contains both static and dynamic behavioural features of Android applications, labeled as benign or ransomware. Before model training, comprehensive preprocessing was performed to ensure data quality and reliability. Duplicate and irrelevant attributes were removed, and the dataset was checked to confirm the absence of missing values. Binary and categorical features were converted into numerical format to make them suitable for machine learning algorithms. Furthermore, numerical features were normalized to enhance training efficiency and stability. All ransomware families were consolidated into a single

ransomware category, resulting in a binary classification problem (benign vs ransomware).

The XGBoost classifier was selected as the core model due to its strong performance in classification tasks, robustness against overfitting, and suitability for feature-based malware detection. Initially, the model was trained using default parameters to establish a baseline performance. To enhance predictive capability, a two-stage hyperparameter tuning process was conducted. In the first phase, key parameters such as max_depth, learning_rate, and n_estimators were optimized. In the second phase, further refinements were made by adjusting subsample, colsample_bytree, and gamma values. This iterative tuning process ensured balanced improvement across accuracy and generalization performance.

To simulate real-time detection, a streaming pipeline was developed using Apache Kafka. Dataset rows were transmitted sequentially through a producer module into Kafka topics, from where a consumer module retrieved the data and applied the trained XGBoost model for immediate prediction. Docker was used to containerize the services, while Zookeeper managed Kafka coordination to ensure stable operation. The system was evaluated using standard performance metrics including Accuracy, Precision, Recall, F1-score, and latency measured in milliseconds, enabling assessment of both predictive power and real-time responsiveness.

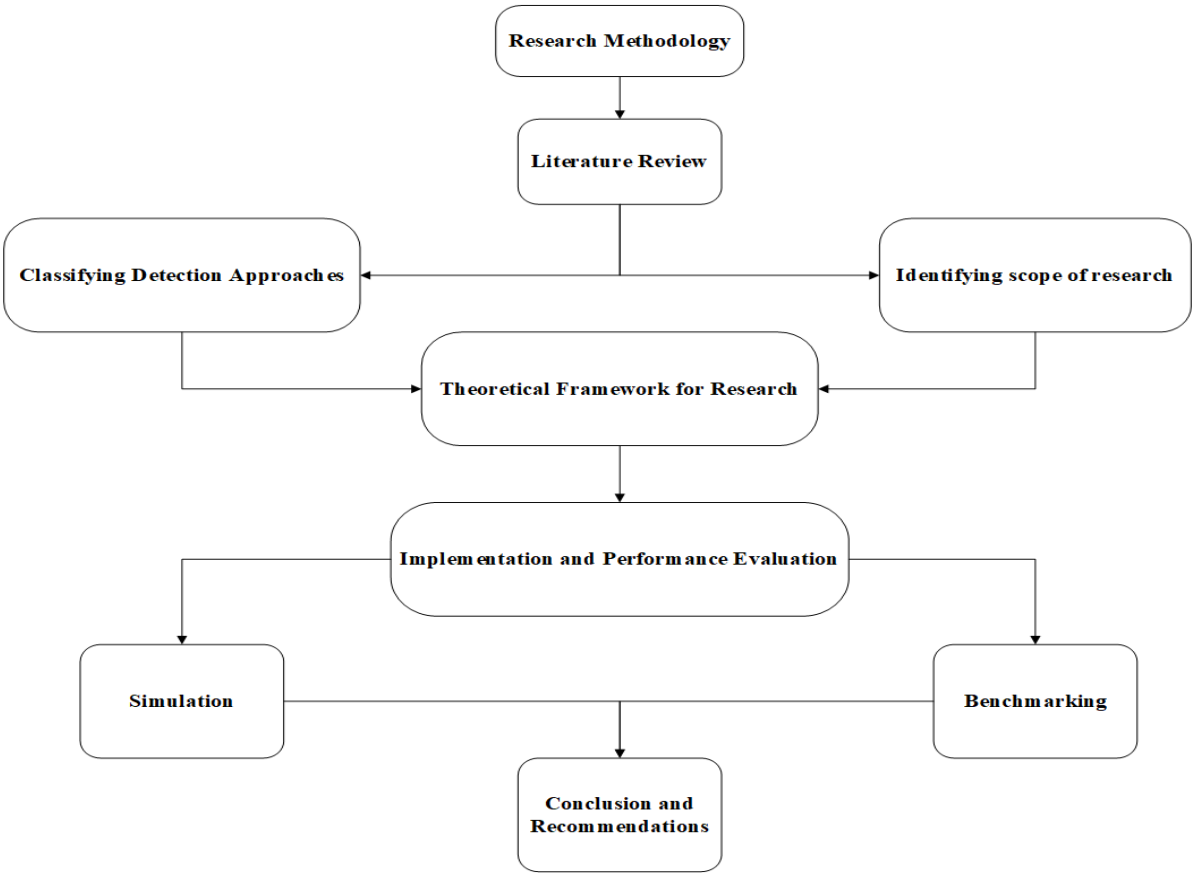


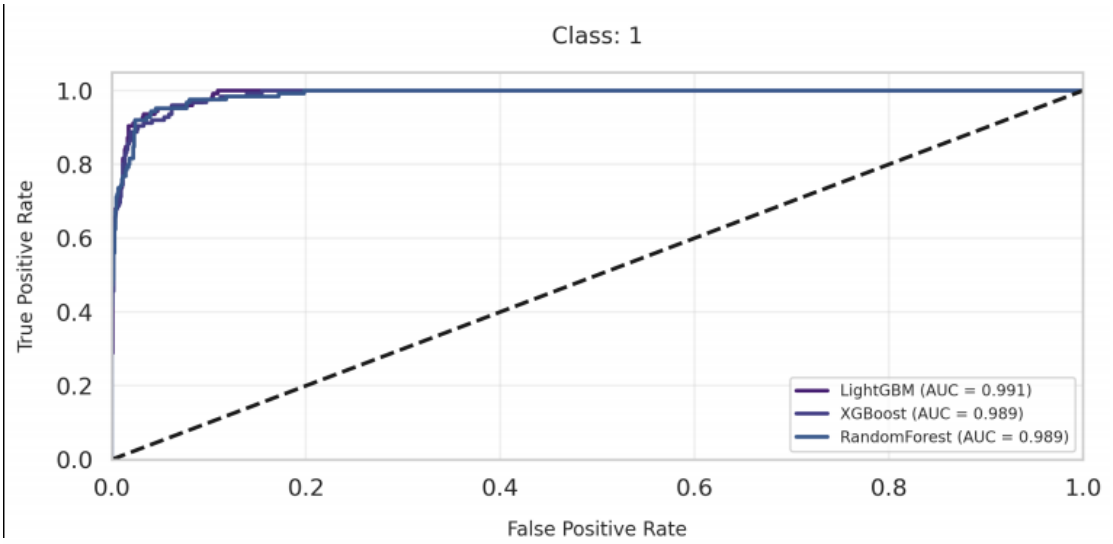
Figure 2: Research Methodology Framework

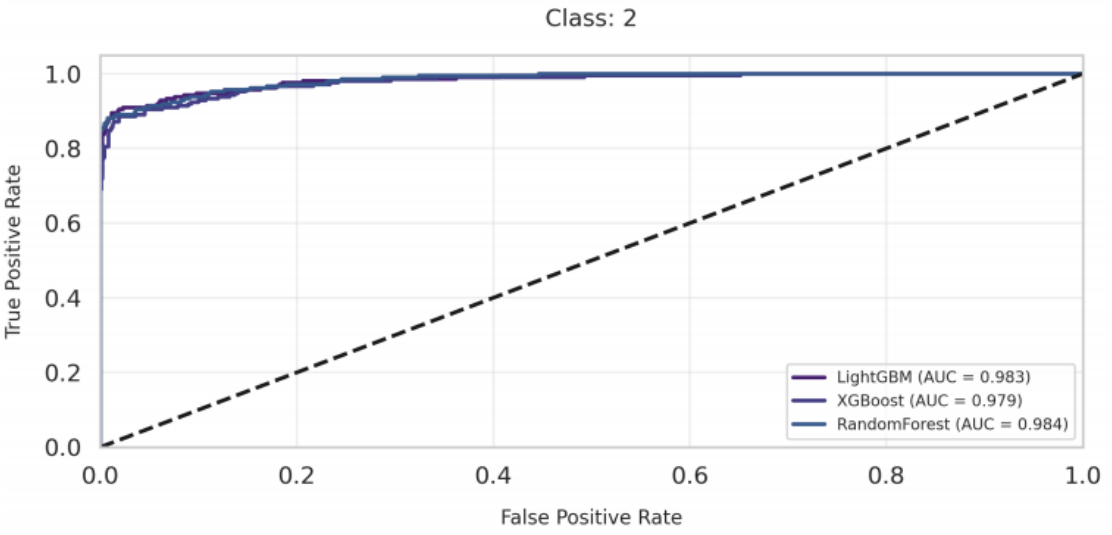
5) Results

The experimental results demonstrate that the proposed system achieved strong performance both in offline training and real-time deployment. The baseline XGBoost model initially achieved an accuracy of 95.00% on the CICMalDroid 2020 dataset, indicating solid predictive capability even before optimization. After the first phase of hyperparameter tuning, the accuracy improved to 96.78%, reflecting the effectiveness of parameter adjustment in enhancing model performance. Following the second tuning phase, the final optimized model achieved an accuracy of 97.42% with an F1-score of 97.35%, confirming balanced precision and recall in ransomware detection.

In the real-time streaming environment, the system maintained high performance while demonstrating low latency. The average model prediction time was approximately 35 milliseconds, while Kafka queue handling introduced an additional 18 milliseconds. The total end-to-end latency was therefore around 53 milliseconds.

These results indicate that the proposed framework successfully combines high detection accuracy with real-time operational capability. Unlike traditional offline approaches, the system proves suitable for time-sensitive ransomware detection scenarios where rapid response is essential to prevent damage.





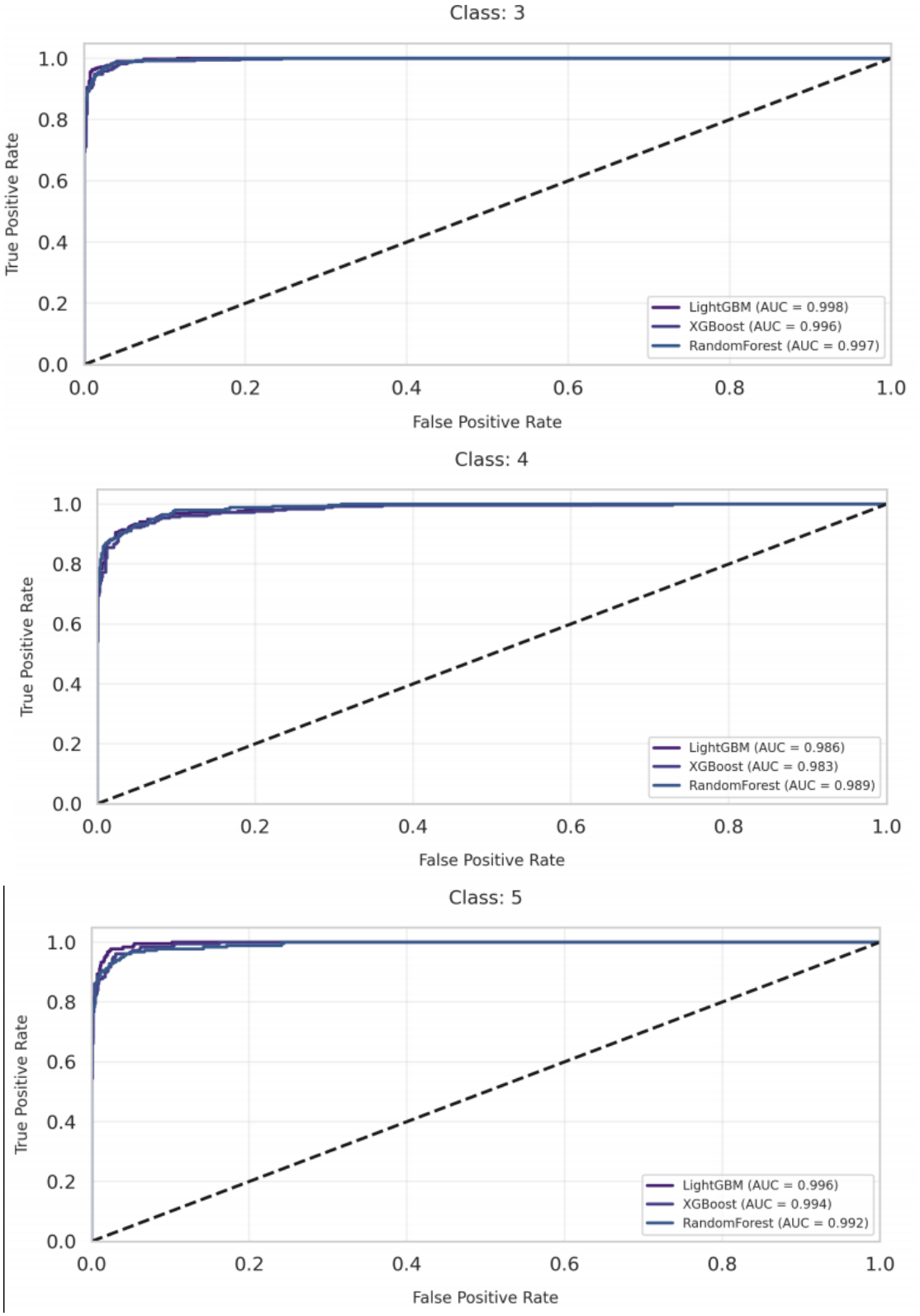


Figure 3: ROC Curve Comparison Across All Classes Using LightGBM, XGBoost, and RandomForest

6) Conclusion

This project developed a real-time Android ransomware detection system using a tuned XGBoost model integrated with Apache Kafka and Docker. The final model achieved 97.42% accuracy while maintaining ~53 ms end-to-end latency, supporting real-time use. The system bridges the common gap between high offline accuracy and practical real-time deployment. Future improvements can include testing on real Android devices and handling concept drift as ransomware evolves.