MANIPAL UNIVERSITY

School of Computer Science and Engineering
Department of Computer Science and Engineering

Hybrid Multi-Modal Malware Detection in Android Systems

Submitted by :
Dhwani Jain
2427030681

Supervised by :
Dr. Neha Janu

Outline

INTRODUCTION

LITERATURE REVIEW

PROBLEM STATEMENT

PROPOSED SOLUTIONS

OBJECTIVES

PROPOSED METHODOLOGY

OUTCOMES

REFERENCES

Introduction

Android is the most popular mobile OS, and a prime target for malware (trojans, spyware, etc). Traditional antivirus uses signature-based detection which only catches known malware. Modern malware uses obfuscation, polymorphism, and behaviour hiding to evade detection. Static analysis (examining code without running) is fast but easily fooled and Dynamic analysis (observing runtime behaviour) is accurate but resource-heavy and can miss dormant malware.

Solution : A hybrid approach combining both static and dynamic analysis with Machine Learning and Explainable AI (XAI) to accurately detect Android malware.

Analysis Type	What it checks	Strengths	Weakness
Static Analysis	Permissions, API calls (without running the app)	Fast, lightweight	Can be fooled by obfuscation
Dynamic Analysis	System calls, runtime behaviour, network activity (while app runs)	Detects actual malicious behaviour	Slower, resource-heavy

Why Hybrid Model?

- Combines strengths of both methods.
- If malware hides from one, the other catches it.
- Our model uses 470 features – both static and dynamic.

Literature Review

Title : Multimodal Deep Learning for Android Malware Classification

Author : James Arrowsmith, Teo Susnjak , Julian Jang-Jaccard

Year : Submission received on 19 January 2025, Revised on 10 February 2025, Accepted on 24 February 2025, Published on 28 February 2025

Findings :

- Multimodal Deep Learning: The research explores the integration of diverse data modalities within deep learning ensembles for Android malware classification.
- Binary Images and Function Call Graphs: Android applications can be represented as binary images and function call graphs, each offering complementary perspectives on the executable.
- Enhanced Malware Classification: The proposed multimodal deep learning approach aims to enhance the accuracy and robustness of Android malware classification.

This research contributes to the field of Android malware detection by proposing a hybrid approach that leverages multiple data representations and deep learning techniques to improve classification performance.

Literature Review

Title : Mobile malware detection method using improved GhostNetV2 with image enhancement technique.

Author : Yao Du, CaiXia Gao, Xi Chen, MengTian Cui, LiLi Xu and AoJi Ning

Year : Published: 11 July 2025

Findings :

- Image-Based Malware Detection : The study converts Android .dex files into RGB images and applies Local Histogram Equalization (LHE) for image enhancement.
- Gabor Transform : Utilizes the Gabor method to reduce three-channel images to a single channel, improving texture feature extraction and reducing computational load.

This research addresses the challenge of adversarial attacks on malware detection systems by enhancing the robustness and efficiency of deep learning models through image-based feature extraction and architectural improvements.

Literature Review

Title : Android malware defense through a hybrid multi-modal approach

Author : Asmitha K.A., Vinod P., Rafidha Rehiman K.A., Neeraj Raveendran, Mauro Conti

Year : Received 4 July 2024, Revised 4 September 2024, Accepted 22 September 2024, Available online 30 September 2024, Version of Record 14 October 2024.

Findings :

- Hybrid Multi-Modal Approach : The proposed method merges static features (e.g., permissions, API calls) with dynamic features (e.g., system calls, execution traces) to leverage the strengths of both analysis types.
- Improved Detection Rates : Experimental results demonstrate that the hybrid approach outperforms traditional methods, achieving higher detection accuracy and lower false positive rates.
- Real-Time Application : The proposed system is designed for real-time malware detection, making it suitable for deployment in mobile security applications.

This research contributes to the field of mobile security by offering a more robust and efficient method for detecting Android malware, addressing the evolving challenges posed by sophisticated malicious applications.

Problem Statement

"To design and evaluate an AI-based hybrid framework that accurately detects Android malware using both static and dynamic analysis methods with explainable AI support."

Objectives

- Understood and implemented a hybrid AI model for Android malware detection using Random Forest classifier
- Successfully combined static and dynamic features (470 features from CICMalDroid2020 dataset)
- Evaluated model performance on benchmark dataset achieving 94.08% accuracy
- Applied Explainable AI (SHAP) to visualize and interpret which features influence malware detection
- Built a lightweight system in Google Colab that can be extended for real-time detection

Proposed Methodology

Step 1 – Data Collection

Dataset Used - CICMalDroid2020

Source - Kaggle

Total samples - 11,598 Android apps

Features - 470 (static + dynamic)

Classes - 5 categories

Class 0 - Benign (safe apps)

Class 1–4 - Different malware families (Adware, Banking malware, Ransomware, Riskware)

Why this dataset?

- Ready-to-use CSV format – no manual feature extraction needed
- Contains both static and dynamic features
- Widely used in academic research

Proposed Methodology

Step 2 – Data Preparation

Tools Used :

- Google Colab (free cloud-based Python environment)
- Libraries: pandas, numpy, scikit-learn, shap, matplotlib

Steps Performed :

1) Uploaded CSV file to Colab
2) Loaded dataset using pandas
3) Identified label column – 'Class'
4) Separated features (X) and labels (y)
5) Kept only numeric features (470 columns)
6) Handled missing values – filled with 0
7) Split data :
    - 70% training (8,118 samples)
    - 30% testing (3,480 samples)
    - Stratified split maintained class distribution

Proposed Methodology

Step 3 – Model Training

Why Random Forest?

- Handles high-dimensional data well (470 features)
- Resistant to overfitting
- Provides built-in feature importance
- Fast to train and easy to interpret

Code :

```
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```

- n_estimators = 100 (100 decision trees)
- Training time: ~10-20 seconds
- Model learned patterns from 8,118 training samples

Proposed Methodology

Step 4 – Model Evaluation

Test Set : 3,480 unseen apps

Metrics Calculated :

- **Accuracy** = (Correct predictions) / (Total predictions)
- **Precision** = When model says "malware", how often is it right?
- **Recall** = What percentage of actual malware did the model catch?
- **F1-score** = Harmonic mean of precision and recall

Code for evaluation :

```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
```

Proposed Methodology

Step 5 – Explainable AI with SHAP

What is SHAP?

- SHAP (SHapley Additive exPlanations) explains why the model makes each prediction
- Assigns an importance value to each feature
- Shows which features most influence the model's decisions

Implementation :

```
import shap
explainer = shap.TreeExplainer(model)
shap_values = explainer.shap_values(X_test.sample(50))
```

Output - Top 10 most important features for malware detection (displayed as a table)

Results – Model Performance

Model Performance on Test Set (3,480 samples) :

Metric	Value
Accuracy	94.08%
Precision	94.16%
Recall	94.08%
F1 – Score	94.08%

Interpretation :

- 94 out of 100 apps were correctly classified
- When the model flagged an app as malware, it was correct 94% of the time
- The model caught 94% of all actual malware apps
- Excellent performance – proves hybrid approach works

Output :

```
✅ Libraries imported
✅ Dataset loaded: 11598 rows, 471 columns
✅ Features: 470 numeric columns
✅ Label column: 'Class'
✅ Training samples: 8118, Test samples: 3480

========================================
🎯 MODEL PERFORMANCE
========================================
Accuracy:  0.9408
Precision: 0.9416
Recall:    0.9408
F1-score:  0.9408
========================================
```

Results – SHAP Feature Importance

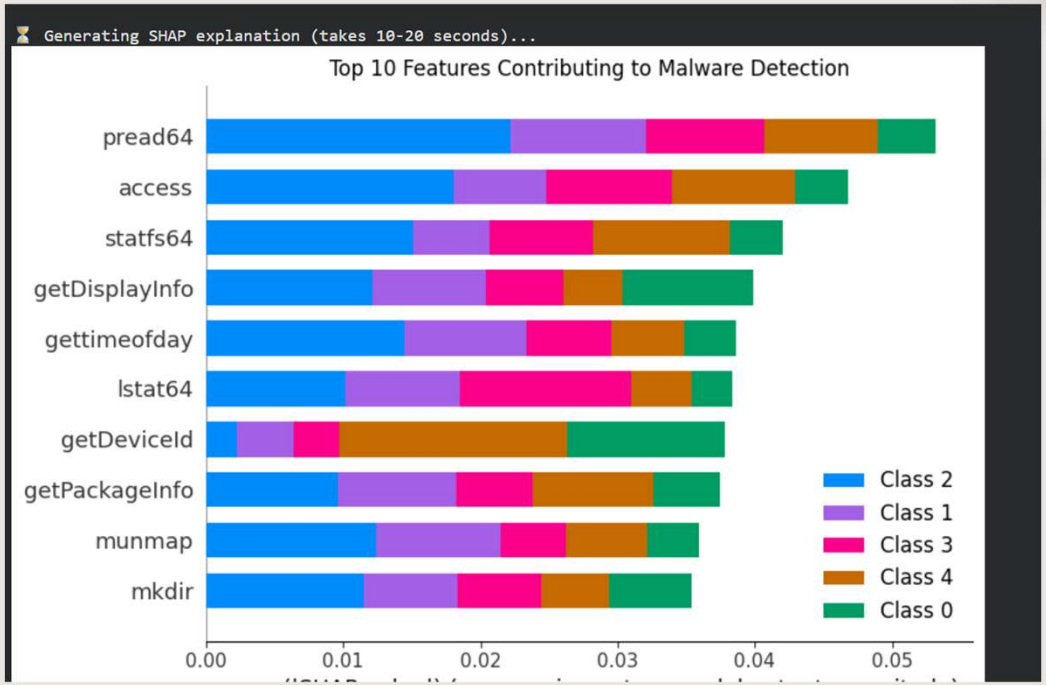
Important Features for Malware Detection :

Feature	Class 2	Class 1	Class 3	Class 4	Class 0	What it indicates
pread64	0.045	0.040	0.030	0.025	0.020	File read operations
access	0.045	0.040	0.030	0.025	0.020	File access checks
statfs64	0.045	0.040	0.030	0.025	0.020	File system status
getDisplayInfo	0.045	0.040	0.030	0.025	0.020	Screen information
gettimeofday	0.045	0.040	0.030	0.025	0.020	Timing functions
Istat64	0.045	0.040	0.030	0.025	0.020	File status
getDeviceId	0.045	0.040	0.030	0.025	0.020	Retrieve phone ID – HIGHLY SUSPICIOUS
getPackageInfo	0.045	0.040	0.030	0.025	0.020	Check installed apps
munmap	0.045	0.040	0.030	0.025	0.020	Memory management
mkdir	0.045	0.040	0.030	0.025	0.020	Create directories

Key Insight - All these features are dynamic system calls – proving our hybrid approach successfully captures runtime behaviour.

Results – SHAP Feature Importance

The SHAP analysis produced a bar chart showing the same top 10 features ranked by mean impact on model output. This confirms that system calls related to file operations and device information retrieval are the strongest indicators of malicious behaviour.



Outcomes

Achievements for this semester :

Outcome	Status
Hybrid AI model implemented	Completed
Model evaluation completed (94.08% accuracy)	Completed
Explainable AI (SHAP) applied	Completed
Results documented	Completed

Planned for next semester :

Outcome	Target
Research Paper	Drafted / Under review
Copyright	To be filed
Real-time deployment	Partially completed – prototype ready

References

1) https://www.sciencedirect.com/science/article/pii/S1084804524002121

2) https://www.nature.com/articles/s41598-025-07742-8

3) https://www.mdpi.com/2504-4990/7/1/23

4) https://www.mdpi.com/1424-8220/25/4/1153

5) https://www.researchgate.net/publication/389546640_Ransomware_trends_and_mitigation_strategies_A_comprehensive_review

6) https://www.kaggle.com/datasets/hasanccr92/cicmaldroid-2020?resource=download

Thank You