Sakshi_s_Paper (2).pdf

Delhi Technological University

Document Details

Submission ID
trn:oid:::27535:97459274

Submission Date
May 24, 2025, 8:30 AM GMT+5:30

Download Date
May 24, 2025, 8:31 AM GMT+5:30

File Name
Sakshi_s_Paper (2).pdf

File Size
321.5 KB

31 Pages

6,866 Words

41,821 Characters

9% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text
- Small Matches (less than 8 words)

Match Groups

44 Not Cited or Quoted 9%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

7% Internet sources
4% Publications
6% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- **44** Not Cited or Quoted 9%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

- 7% Internet sources
- 4% Publications
- 6% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	dspace.dtu.ac.in:8080 2%
2	Submitted works	Jawaharlal Nehru Technological University on 2025-05-13 <1%
3	Internet	www.trademarkelite.com <1%
4	Submitted works	University of Sussex on 2025-03-21 <1%
5	Publication	Sandeep Sharma, Prachi, Rita Chhikara, Kavita Khanna. "chapter 12 Hybrid Featu... <1%
6	Submitted works	The Robert Gordon University on 2013-10-08 <1%
7	Submitted works	Liverpool John Moores University on 2024-11-11 <1%
8	Internet	wiki.expertiza.ncsu.edu <1%
9	Submitted works	University of Bradford on 2024-12-16 <1%
10	Submitted works	Carnegie Mellon University on 2024-11-18 <1%

11 Internet

epdf.pub <1%

12 Submitted works

De Montfort University on 2024-04-21 <1%

13 Submitted works

National College of Ireland on 2023-04-16 <1%

14 Submitted works

University of Glasgow on 2024-08-08 <1%

15 Internet

learnmetaheuristics.blogspot.com <1%

16 Internet

www.dspace.dtu.ac.in:8080 <1%

17 Publication

Andrea Augello, Alessandra De Paola, Giuseppe Lo Re. "Hybrid Multilevel Detectio... <1%

18 Publication

Kuncup Iswandy, Andreas Koenig. "Feature-Level Fusion by Multi-Objective Binar... <1%

19 Internet

doczz.net <1%

20 Internet

ijns.jalaxy.com.tw <1%

21 Internet

jaysinha.me <1%

22 Internet

jsaer.com <1%

23 Internet

wseas.com <1%

24 Publication

Ally S. Nyamawe, Mohamedi M. Mjahidi, Noe E. Nnko, Salim A. Diwani, Godbless G... <1%

25 Submitted works
Cranfield University on 2025-01-06 <1%

26 Publication
Faitouri A. Aboaoja, Anazida Zainal, Fuad A. Ghaleb, Norah Saleh Alghamdi, Faisal... <1%

27 Publication
Sai Kiran Oruganti, Dimitrios A Karras, Srinesh Singh Thakur, Janapati Krishna Ch... <1%

28 Submitted works
University of Hertfordshire on 2023-08-27 <1%

29 Submitted works
University of Hertfordshire on 2024-09-02 <1%

30 Submitted works
University of Pretoria on 2006-07-30 <1%

31 Submitted works
University of Reading on 2024-09-12 <1%

32 Submitted works
University of Southampton on 2007-02-01 <1%

33 Publication
Youssef Baddi, Mohammed Amin Almaiah, Omar Almomani, Yassine Maleh. "The ... <1%

34 Internet
al-kindipublishers.org <1%

35 Internet
bradscholars.brad.ac.uk <1%

36 Internet
ijcem.in <1%

37 Internet
www.jisem-journal.com <1%

38 Internet
www.nature.com <1%

39 Internet

www.tnsroindia.org.in <1%

Optimizing Android Malware Detection via Hybrid Feature Selection: A Study on Z-Test, MIFS & PSO

A DISSERTATION

Submitted in partial fulfillment of the requirements
for the award of the degree

MASTER OF SCIENCE (M.Sc.)
in
MATHEMATICS

Submitted by

Sakshi Punia
(23/MSCMAT/39)

Under the supervision of

Dr. Anshul Arora

DEPARTMENT OF APPLIED MATHEMATICS
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

MAY, 2025

Candidate's Declaration

I, Sakshi Punia, Roll No. 23/MSCMAT/39 student of Master in Science (Mathematics), state that the dissertation I turned in to Delhi Technological University's Department of Applied Mathematics, with the title **Optimizing Android Malware Detection via Hybrid Feature Selection: A Study on Z-Test, MIFS & PSO** is completely authentic and free of any copies of other sources without the required citation. Part of the prerequisites for earning a Master of Science in Mathematics degree are met by this.

Place: Delhi
Date: May 26, 2025

Sakshi Punia
23/MSCMAT/39

Certificate

I certify that the project dissertation **Optimizing Android Malware Detection via Hybrid Feature Selection: A Study on Z-Test, MIFS & PSO** which was turned in by Sakshi Punia, Roll No. 23/MSCMAT/39 of the Department of Applied Mathematics at Delhi Technological University, Delhi, as a partial fulfilment of the requirements for the award of the Masters of Science in Mathematics degree, is a record of the work completed by the student under my guidance. To the best of my knowledge, neither this university nor any other has accepted this work in whole or in part for a degree or diploma.

Place: Delhi
Date: May 26, 2025

Dr. Anshul Arora
Supervisor

Acknowledgement

My supervisor, Dr. Anshul Arora of the Department of Applied Mathematics at Delhi Technological University, has my sincere gratitude for his meticulous guidance, profound expertise, constructive criticism, attentive listening, and amiable demeanor have been invaluable throughout the process of composing this report. I am eternally grateful for his benevolent and supportive approach, as well as his perceptive counsel, which played an important role in the successful culmination of my dissertation. Furthermore, I would like to express my appreciation to all my classmates who have played a pivotal role in aiding me to complete this endeavor by offering assistance and facilitating the exchange of pertinent information.

Sakshi Punia
23/MSCMAT/39

Abstract

The increasing prevalence of Android malware poses significant security risks, necessitating efficient detection techniques. With Android being the mobile operating system with the largest user base, it has become a key focus for harmful apps that take advantage of weaknesses to breach user privacy and device functionality. The rapid evolution and obfuscation of malware further challenge traditional detection approaches, making advanced detection strategies crucial.

This study explores Android malware detection using binary data while optimizing feature selection through Z-test, Mutual Information Feature Selection (MIFS), and Particle Swarm Optimization (PSO). Binary representation provides a structured and compact format, enabling consistent feature extraction from Android application packages (APKs). The three feature selection techniques employed in this study aim to eliminate irrelevant or redundant features, thereby enhancing model efficiency and detection accuracy. Z-test helps in identifying statistically significant features, MIFS evaluates the relevance of features based on mutual information, and PSO searches for an optimal feature subset using a population-based heuristic.

We analyze malware detection performance across three feature sets—hardware, intents, and permissions—comparing results with and without feature selection. These feature categories are commonly used in static analysis and provide critical insights into app behavior, making them valuable for classification tasks. The comparison helps assess how much the selected features contribute to classification performance while reducing computational burden.

Experimental findings reveal that applying feature selection significantly reduces the number of features (from 297 to as few as 55) while maintaining or improving classification accuracy. This reduction not only minimizes overfitting but also speeds up the training process. The model with the best performance achieved 97% accuracy, demonstrating that feature selection enhances malware detection while reducing computational complexity. The study confirms that thoughtful feature selection is essential for building lightweight, high-performing malware classifiers suited for real-world deployment.

Contents

List of Tables

List of Figures

Introduction

Smartphones are now an indispensable aspect of contemporary life, transforming communication, business, and access to basic services like banking, healthcare, and government services. The increased dependence on mobile technologies has reshaped digital interaction patterns, making smartphones a central tool for modern living. Among mobile operating systems, Android stands out due to its versatility, user base, and developer-friendly environment.

Android, with more than 70% of the world's market share as of 2023, is the leading mobile platform, thanks to its open-source platform and popularity among device makers. Its openness encourages innovation but also exposes the system to security vulnerabilities. Yet this extensive usage also renders Android a top target for cyber attacks, especially malware, which remains constantly evolving in sophistication and volume. As malicious actors find new ways to exploit system loopholes, the threat landscape becomes increasingly difficult to monitor and control using conventional methods.

1.1 Background

Malware is an overarching term used to describe malicious software that has been created to take advantage of vulnerabilities, disable system operations, or capture sensitive information. It is a significant threat to Android users, impacting both individuals and organizations. Android users are under threat from all types of malware, ranging from trojans, ransomware, and spyware to adware, each designed to exploit the system in specific ways. These attacks compromise user privacy, device performance, and sometimes result in irreversible data loss.

Signature-based antivirus technologies are unable to cope with the fast pace of new malware strain development, as malware developers frequently use obfuscation techniques to avoid detection. Thus, adaptive detection strategies are crucial. In this context, machine learning-driven techniques have emerged as a dominant force in detecting malware by spotting suspicious behavioral activity and separating good from bad apps. These approaches provide the flexibility to identify previously unseen malware by learning patterns from labeled datasets.

Feature selection is an important factor in improving malware detection effectiveness. Selecting meaningful features reduces model complexity and improves generalization. Out of several static features, permissions, intents, and hardware features have demonstrated the ability to differentiate between malicious applications. Permissions, which describe an app's rights to access device capabilities, tend to act as security threat indicators. However, examining all extracted features without filtering causes noise, redundancy, and increased computational cost. This not only affects detection performance but also

slows down real-time response.

Thus, employing feature selection techniques like Z-test, Mutual Information Feature Selection (MIFS), and Particle Swarm Optimization (PSO) is beneficial in reducing the most relevant features, enhancing classification accuracy, and lessening processing time. These techniques help in selecting statistically and informationally significant attributes, and in navigating large feature spaces efficiently. This paper investigates the impact of these feature selection methods in improving Android malicious software identification models such that there is an enhanced and scalable security framework suited for evolving threats.

1.2 Motivation

The rapid expansion of Android applications has made mobile security a growing concern, with malware threats evolving at an unprecedented rate. Traditional signature-based detection techniques are becoming insufficient to combat advanced malware variants that use obfuscation and evasion techniques. As a result, machine learning-based detection systems have become prominent for their capacity to recognize patterns and anomalies within application behavior. However, a major challenge in such approaches is handling high-dimensional data, where an excessive number of features can lead to overfitting, increased computation time, and reduced model interpretability.

To overcome these challenges, this study investigates a feature selection methodology that refines the selection process by applying Z-test, Mutual Information Feature Selection (MIFS), and Particle Swarm Optimization (PSO). By systematically filtering out irrelevant and redundant features, this strategy focuses on refining the accuracy of malware classification while ensuring the model remains computationally efficient. The primary objective is to create a robust and scalable malware detection system that can effectively distinguish between benign and malicious applications, contributing to improved Android security solutions.

Furthermore, selecting the most relevant features not only improves detection accuracy but also enhances the model's interpretability, allowing security analysts to better understand the key indicators of malware. A refined feature set ensures that the classification process remains efficient, reducing unnecessary complexity and making the approach more adaptable to new and evolving threats. By focusing on a lightweight yet effective detection mechanism, this study attempts to address the gap between high detection performance and real-world feasibility, ultimately contributing to the development of robust mobile security solutions.

1.3 Contribution

This paper presents the following significant contributions:

- *Introduces a new feature selection methodology* by systematically applying Z-Test, Mutual Information Feature Selection(MIFS), and Particle Swarm Optimization(PSO) to refine malware detection features from Android applications. This multi-stage approach enhances detection accuracy while reducing computational overhead.

- *Comprehensive feature extraction and analysis*: The study utilizes a dataset comprising static characteristics of Android applications, including permissions, hardware components, and intents. The original dataset contained 297 features, which were reduced to 55 optimal features through MIFS and PSO, ensuring an efficient and lightweight detection system. This analysis is conducted on an exhaustive dataset of over 111,000 Android applications.

- *Demonstrates the effectiveness of feature selection*: The study evaluates malware classification performance before and after applying feature selection techniques across multiple feature groups—hardware, permissions, and intents. The results indicate that removing redundant and irrelevant features leads to improved classification accuracy while maintaining a reduced feature set. The final model achieves a peak accuracy of 97.2% with a minimal feature set, demonstrating the efficiency of the proposed methodology.

- *Provides an optimized malware detection framework* that balances three key aspects of feature selection: statistical significance (Z-Test), feature relevance (MIFS), and optimization-driven reduction (PSO). This approach ensures that selected features are not only relevant but also contribute to efficient model training and deployment.

- *Demonstrates the impact of feature selection on the comprehensive dataset*, (Hardware + Permissions + Intents), reducing features from 297 to 55 while maintaining high classification accuracy.

- *Enhances model interpretability and efficiency*: By reducing the number of features while maintaining high accuracy, this study contributes to the formation of a lightweight malware detection system which is both scalable and interpretable, making it viable for real-world deployment in mobile security applications.

- *Establishes the applicability of feature selection in malware detection* by systematically comparing results with and without feature selection, proving that reducing feature dimensionality does not compromise detection effectiveness but rather enhances computational efficiency and classification performance.

1.3.1 Why This is a Novel Approach

What makes this approach novel is its structured, multi-stage feature selection process combining three independent yet complementary techniques: Z-Test (statistical filtering), MIFS (relevance-based selection), and PSO (optimization-based refinement). This tiered framework ensures that features are not only statistically significant and relevant to the target variable but also optimized for model performance through an intelligent search strategy.

While each method has been used in isolation in past research, this study is the first to integrate them into a cohesive pipeline for Android malware detection. The combination leverages the strengths of each technique—Z-Test for early-stage noise removal, MIFS for capturing mutual dependencies between features and class labels, and PSO for fine-tuning the selected feature subset to yield the highest classification accuracy. This layered methodology results in a compact yet powerful feature set that improves both

the efficiency and scalability of the detection model. By applying this approach on binary static features such as permissions, intents, and hardware data, the study presents a lightweight, generalized, and practical solution suited for real-time malware analysis.

1.4 Thesis Structure

The organization of the remaining thesis is as follows:

- An overview of the suggested feature selection methods namely Z-TEst, MIFS and PSO is given in Chapter 2, along with an introduction to the fundamental ideas of malware and important features like permissions, intents, and hardware components.
- A review of related work is given in Chapter 3 which talks about the existing research on Android malware detection and feature selection techniques.
- The dataset and the pre-processing procedures used to extract and prepare features from Android applications are covered in Chapter 4.
- The suggested methodology is described in Chapter 5, which also goes into detail about the machine learning classifiers used for detection and each step of the hybrid feature selection pipeline.
- A thorough analysis of the experimental findings is given in Chapter 6, which also compares different feature combinations and highlights important model performance findings.
- A conclusion provided in Chapter 7, which brings the thesis to a close.
- The future scope of the work and its possible Societal Impacts are examined in Chapter 8

Chapter 2

Before proceeding further, it is imperative to establish a foundational understanding of the fundamental concepts essential for comprehending the ensuing work. These concepts include:

1. What is Malware and its types ?
2. What are Permissions, Intents, and Hardware Features in Android?
3. What is Feature Selection and Why is it Important?
4. What are the Z-Test, MIFS & PSO ?

2.1 Malware and Types?

Malware, an abbreviation for malicious software, denotes any program purposely created to interfere with, harm, or unlawfully access computer systems or data. In the context of Android, malware includes threats like trojans, ransomware, spyware, and adware, which compromise user privacy, steal sensitive information, or harm device functionality.

Types of Android Malware

- **Trojans:** Disguised as legitimate apps, they perform malicious actions in the background, such as stealing data or downloading additional malware.
- **Ransomware:** Encrypts or locks users out of their devices or data, demanding payment for restoration
- **Spyware:** Silently monitors user activity, including keystrokes, location, and personal information, and sends it to attackers.
- **Adware:** Displays intrusive ads, often slowing down the device and collecting user data without consent.
- **Worms:** Self-replicating malware that spreads through networks or messaging apps without user interaction.
- **Backdoors:** Provide attackers with remote control over the infected device, allowing unauthorized access to system resources.

2.2 Android Features: Permissions, Intents, and Hardware

Permissions

These define the access rights an app requests to use device resources, such as location or contacts. Malicious apps often request excessive permissions to exploit sensitive data.

Intents

Intents enable communication between different components of an app. Certain intents can indicate suspicious behavior when used to trigger unauthorized actions.

Hardware Components

This refers to the physical device features accessed by an app, like the camera or GPS. Malware may misuse hardware access for spying or data theft.

2.3 Feature Selection and it's Importance

Feature selection is the technique used to determine and retain the most informative and relevant features from a larger collection of input parameters for use in building predictive models. In the context of high-dimensional datasets, particularly those with hundreds or thousands of features, not all variables contribute equally to model performance. Many may be redundant, irrelevant, or even introduce noise, which can hinder both the accuracy and efficiency of the model. After selecting a subset of meaningful features, the model becomes more focused and is better equipped to capture the underlying patterns in the data.

The significance of feature selection lies in its multiple benefits. It plays a key role in mitigating overfitting via feature elimination that cause the model to learn noise rather than true signal. This leads to better generalization on unseen data. Additionally, it lowers the computational cost by simplifying the data by reducing dimensions of the data, which is especially crucial when working with resource-constrained environments or large-scale systems. Another significant advantage is the improvement in model interpretability, which is particularly important in security applications where understanding why a sample is flagged as malicious can support further forensic analysis.

In the domain of Android malware detection, feature selection is essential for identifying those attributes—such as permissions, API calls, or behavioral signatures—that most reliably differentiate between benign and malicious applications. This targeted approach not only accelerates the detection process but also increases its reliability by focusing on the most indicative markers of harmful behavior. As malware techniques evolve, robust feature selection allows detection systems to remain adaptable and responsive to new threats while minimizing false positives.

2.4 Z-Test, MIFS & PSO

Z-Test, Mutual Information Feature Selection (MIFS), and Particle Swarm Optimization(PSO) are feature selection techniques used to improve machine learning model performance by selecting the most relevant features.

- **Z-Test:** A statistical test that measures the significance of each feature by comparing means between classes. It helps filter out features that do not show a strong statistical difference between malicious and benign samples
- **Mutual Information Feature Selection (MIFS):** A relevance-based method that evaluates the dependency between features and class labels. MIFS selects features that offer the highest informational value about the target variable while minimizing redundancy.
- **Particle Swarm Optimization(PSO):** An optimization technique modeled after the collective behavior observed in bird flocks or fish schools. PSO searches for the best subset of features by iteratively improving candidate solutions based on a fitness function, such as classification accuracy.

Together, these techniques complement each other by combining statistical significance, information theory, and optimization to effectively cut down feature dimensions with no loss—and possible gain—in detection accuracy.

Related work

Android malware detection has been extensively studied, with numerous approaches leveraging feature selection and machine learning techniques.

Kim and Choi [1] explored Linux kernel-based feature selection for Android malware detection, demonstrating its efficacy in reducing computational complexity. Adriansyah et al. [2] employed ensemble learning and SHAP explainable AI for feature selection, providing insights into the importance of different features. D. J. et al. [3] introduced a multimodal feature selection approach to improve classifier performance in Android malware detection.

Eom et al. [4] analyzed feature selection techniques in combination with Random Forest, highlighting their impact on detection accuracy. Fatima et al. [5] proposed a genetic algorithm-based feature selection method, optimizing feature subsets for improved classification. K. S. J. et al. [6] evaluated permission-based feature selection methods, emphasizing their effectiveness in malware detection.

Khalid and Hussain [7] assessed dynamic analysis features for Android malware categorization, while Guyton et al. [8] performed a comparative analysis of multiple feature selection techniques. Nezhadkamali et al. [9] investigated overlapping static features, demonstrating their role in malware classification. Wang et al. [10] leveraged XGBoost for feature selection and malware detection, reporting significant performance gains.

Dhalaria and Gandotra [11] used chi-square-based feature selection with ensemble learning for enhanced detection accuracy. Sahal et al. [12] focused on mining Android permissions to identify malware patterns. Zhu et al. [13] explored API sequence-based malware detection, showing improvements in classification accuracy. Tarar et al. [14] analyzed machine learning algorithms for Android malware classification, highlighting key features contributing to detection.

Awasthi et al. [15] introduced RFECV-DT, a recursive feature selection approach with decision trees for optimized detection performance. Guyton et al. [16] examined permissions, intents, and API calls for feature selection in malware detection. Nivaashini et al. [17] compared various feature selection methods and machine learning algorithms for permission-based malware classification.

Sharma et al. [18] utilized dynamic analysis and recursive feature elimination (RFE) with artificial neural networks (ANN) for malware detection. Patel [19] provided a comprehensive study on Android malware detection methodologies. Kadir and Peddoju [20] presented a hybrid feature-based malware detection model integrating multiple feature types. Li et al. [21] proposed an approach based on the AndroidManifest file, identifying critical features for malware classification. Sahal et al. [22] explored permission-based feature selection for Android malware detection, highlighting the potential of lightweight features in reducing model complexity. Hadiprakoso et al. [23] proposed a hybrid-based

analysis framework combining static and dynamic techniques to enhance detection efficiency. Lu and Hou [24] introduced a two-layered permission-based detection model, aiming to refine classification through hierarchical feature processing. Park et al. [25] focused on detecting malware by assessing similarity with benign Android applications, emphasizing structural and behavioral patterns. Baghirov [26] presented a comprehensive detection framework that leverages ensemble methods, advanced feature selection, and hyperparameter optimization to achieve high accuracy.

These studies collectively emphasize the significance of applying feature selection in Android malware detection, demonstrating how various techniques can enhance classification accuracy and computational efficiency.

Based on these previous attempts, our work proposes a systematic, multi-step feature selection model that successively combines Z-Test, Mutual Information Feature Selection (MIFS), and PSO. In contrast to previous research where these methods are applied separately, our method consolidates them into an efficient pipeline for better Android malware classification. We also evaluate our model on a large dataset of more than 111,000 apps, making it highly robust and applicable for use in real-world environments.

Dataset and Preprocessing

Our study utilizes a large dataset of over 111,000 Android applications, comprising both benign and malicious samples. The dataset includes three key feature categories:

- **Permissions:** Defines the access levels requested by an application.
- **Hardware:** Represents the physical device features accessed by the application.
- **Intents:** Specifies the communication between different components of an application.

To ensure data quality, preprocessing steps such as duplicate removal and handling of missing values were performed.

4.1 Feature Extraction

The dataset was sourced from Androzoo, a large collection of Android application packages (APKs) curated for security research. The static features were extracted from the *AndroidManifest.xml* file of each application using Apktool, which decompiles APKs into readable formats. The extracted feature sets include:

- Permissions: Binary indicators for 129 distinct permissions.
- Intent Filters: Event-driven components.
- Hardware Features: Device-specific attributes such as GPS, Camera, and Bluetooth.

Each feature was represented as a binary value, where 1 indicates presence and 0 indicates absence.

4.2 Preprocessing Steps

To ensure data quality, the preprocessing steps included:

- Extracting and converting the code into structured feature matrices using Python scripts.
- Assigning labels, where **1 represents malware** and **0 represents benign** samples.
- Normalizing the dataset and splitting it into training and testing sets for assessment.

Methodology

To refine the feature space and enhance classification performance, we applied a sequential feature selection approach using three key methods: Z-Test, Mutual Information Feature Selection (MIFS), and Particle Swarm Optimization (PSO). Each of these techniques contributes uniquely to reducing redundancy and improving feature significance.

5.1 Feature Selection Methods Used

This section outlines a hybrid approach for feature selection in Android malware detection. The process integrates three stages: Z-Test, Mutual Information, and Particle Swarm Optimization (PSO). Each stage progressively refines the feature set to improve classification performance.

1. **Z-Test:** Z-Test is a statistical hypothesis technique applied to ascertain if a feature is significantly different between malware and benign samples. It helps identify features that have a meaningful impact on classification.

   i. **Formula:** For a given binary feature X, the Z-score is computed as:

   $$Z = \frac{p_1 - p_2}{\sqrt{p(1-p)\left(\frac{1}{N_1} + \frac{1}{N_2}\right)}} \tag{5.1}$$

   where:

   - $p_1 = \frac{X_{\text{malware}}}{N_1}$ is the proportion of malware samples having the feature.
   - $p_2 = \frac{X_{\text{benign}}}{N_2}$ is the proportion of benign samples having the feature.
   - $p = \frac{X_{\text{malware}} + X_{\text{benign}}}{N_1 + N_2}$ is the overall proportion.
   - N_1, N_2 are the number of malware and benign samples, respectively.

   ii. **Steps Followed:**

   i. Calculate the presence ratio of each feature in malware and benign classes.
   ii. Compute the Z-score using the formula above.
   iii. Apply a significance threshold to filter out non-discriminative features.

2. **Mutual Information Feature Selection(MIFS):** Mutual Information (MI) quantifies the amount of shared information between a feature and the target class. A high MI value indicates a strong dependency between the feature and the malware/benign label.

i. **Formula:**

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} P(x,y) \log \frac{P(x,y)}{P(x)P(y)} \tag{5.2}$$

where:

- $P(x,y)$ is the joint probability distribution of feature X and class Y.
- $P(x)$ and $P(y)$ are the marginal probability distributions.

ii. **Steps Followed:**

i. Compute the MI value for each feature with respect to the class label.
ii. Rank the features based on MI scores.
iii. Select the top k features with the highest MI values.

3. **Particle Swarm Optimization(PSO):** Particle Swarm Optimization(PSO) is an optimization technique modeled after the collective behavior observed in bird flocks or fish schools. It initializes a population (swarm) of candidate solutions, called particles, which navigate the solution space to find an optimal feature subset based on a defined fitness function. In feature selection, every particle signifies a possible subset of features, and the objective is to identify the subset yielding the highest classification accuracy.

Each particle i in the swarm is defined by two vectors:

- **Position vector** $x_i(t)$ – represents the selected subset of features at iteration t.
- **Velocity vector** $v_i(t)$ – indicates the direction and magnitude of movement in the solution space.

The velocity and position of each particle are updated using the following equations:

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot r_1 \cdot (p_i - x_i(t)) + c_2 \cdot r_2 \cdot (g - x_i(t)) \tag{5.3}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{5.4}$$

Where:

- w is the inertia weight balancing global and local search.
- c_1 and c_2 are acceleration coefficients representing cognitive (personal) and social (global) influences.
- r_1 and r_2 are random values drawn from a uniform distribution in $[0, 1]$.
- p_i is the personal best position of particle i.
- g is the global best position identified by the swarm.

PSO Algorithm Steps for Feature Selection:

(a) **Initialization:** Initialize a swarm of particles with random binary strings, where each bit represents whether a feature is selected (1) or not (0).

(b) **Fitness Evaluation:** Evaluate each particle using a fitness function, typically based on classification accuracy of a machine learning model.

(c) **Update Bests:** Update each particle's personal best p_i and the global best g based on the fitness evaluations.

(d) **Update Velocities and Positions:** Apply Equations 5.3 and 5.4 to adjust each particle's state.

(e) **Binary Transformation:** Apply a sigmoid function to the velocity vector to obtain selection probabilities:

$$S(v_i) = \frac{1}{1 + e^{-v_i}}$$

A feature is selected ($x_i = 1$) if a randomly generated number is less than $S(v_i)$; otherwise, it is not selected ($x_i = 0$).

(f) **Termination:** Repeat the process until convergence or a predefined number of iterations is reached.

This approach balances exploration and exploitation to converge on an optimal feature subset. In this study, PSO is applied after Z-test and Mutual Information steps to fine-tune the feature selection, ensuring a compact and high-performing feature set for malware classification.

This sequential approach ensures that we first remove irrelevant features using Z-Test, retain only the most informative ones using MI, and finally refine the subset using PSO to maximize classification accuracy.

5.2 Hybrid Feature Selection Pipeline

To refine the dataset and enhance classification performance, we employed a three-step feature selection approach:

1. Z-Test for Initial Filtering:
   - The Z-Test was applied to the entire dataset to assess the statistical significance of each feature in distinguishing malware from benign samples.
   - Features with low discriminatory power (high p-value) were removed, resulting in a reduced feature subset for further processing.
2. Mutual Information Feature Selection (MIFS):
   - The features retained from the Z-Test were further evaluated using Mutual Information (MI) to measure their dependency on the class label.
   - Features with the highest MI scores were selected to ensure that only the most relevant ones proceeded to the next stage.
3. Particle Swarm Optimization (PSO):
   - The final refined feature set from MIFS was optimized using PSO, which selected an optimal subset maximizing classification accuracy while minimizing redundancy.

- This step helped achieve a balance between model effectiveness and computational cost.

By following this sequential pipeline, we prioritized exclusively the most relevant and non-redundant features were retained for malware classification.

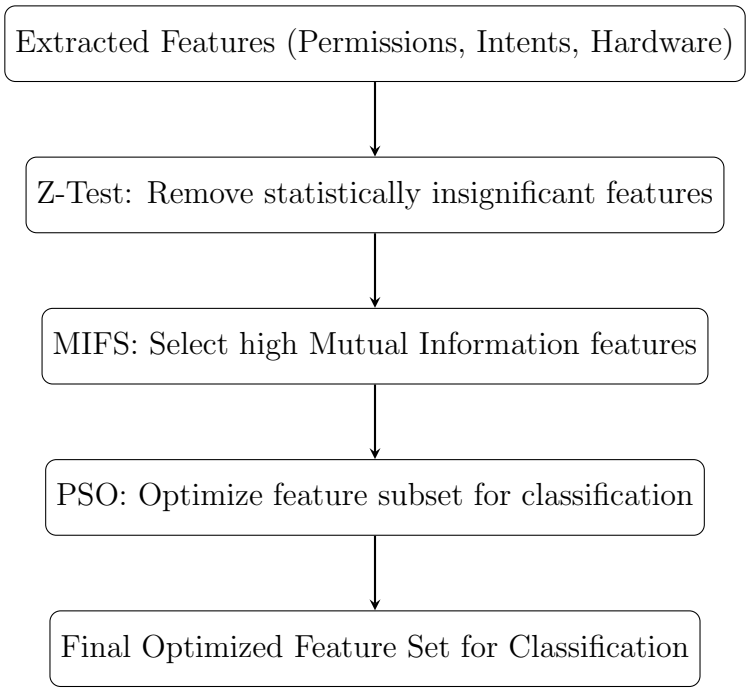

Figure 5.1: Feature Selection Pipeline

5.3 Classification Models

21 To comprehensively evaluate the impact of feature selection, we applied the following classifiers:

- **Decision Tree**
- **Random Forest**
- **Gaussian Naive Bayes**

The classifiers were initially trained on the full dataset without feature selection and later on feature subsets obtained after each selection method (Z-Test, MIFS, and PSO). Additionally, we assessed performance separately on individual feature categories (Permissions, Intent Filters, and Hardware Features) before merging them into a unified dataset for final evaluation. Model effectiveness was measured using accuracy, standard deviation, and confusion matrices.

Results and Discussion

The following section outlines the evaluation of classification accuracy and feature reduction using different feature selection techniques. The main focus is on the **Hardware + Permissions + Intents** dataset, as it provides the most comprehensive analysis. Individual and paired datasets are briefly discussed for comparative insights.

6.1 Feature Ranking Results

The top 10 ranked features from each category—Permissions, Intents, and Hardware Components—are listed in the table below.

TABLE I
TOP 10 RANKED FEATURES BY CATEGORY

Permissions	Intents	Hardware
READ_PHONE_STATE	BOOT_COMPLETED	TOUCHSCREEN
CAMERA	PACKAGE_ADDED	touchscreen.multitouch
READ_SMS	BATTERY_LOW	screen.portrait
SEND_SMS	MESSAGE_RECEIVED	screen.landscape
ACCESS_FINE_ _LOCATION	NOTIFICATION_ _RECEIVED	location.network
MODIFY_AUDIO_ _SETTINGS	CONNECTION	CAMERA
CALL_PHONE	ACTION_RICHPUSH_ _CALLBACK	EXTERNAL_STORAGE
WRITE_CONTACTS	BROADCAST_PACKAGE_ _ADDED	VIBRATE
WRITE_SETTINGS	RECEIVE_BOOT_ _COMPLETED	TELEPHONY
KILL_BACKGROUND_ _PROCESSES	MY_PACKAGE_ _REPLACED	BLUETOOTH

Figure 6.1: Top 10 Ranked Features by Category

6.2 Detecting Results with Merged Features

The dataset initially contained 297 features, which were reduced using three different feature selection techniques: Z-Test, MIFS, and PSO. The classification performance

before and after feature selection is summarized below.

6.2.1 Without Feature Selection

The baseline classification results without feature selection are as follows:

Model	Mean Accuracy
Naïve Bayes	90.78%
Decision Tree	96.93%
Random Forest	97.74%

Table 6.1: Classification accuracy without feature selection

6.2.2 With Feature Selection

(a) **Z-Test:** Reduced the features to 281.

Model	Mean Accuracy
Naïve Bayes	89.02%
Decision Tree	96.50%
Random Forest	97.45%

Table 6.2: Z-Test results on merged dataset

(b) **MIFS:** Reduced the features to 100.

Model	Mean Accuracy
Naïve Bayes	87.75%
Decision Tree	95.93%
Random Forest	97.11%

Table 6.3: MIFS results on merged dataset

(c) **PSO:** Provided the most significant feature reduction to 55 features.

Model	Accuracy
Naïve Bayes	88.75%
Decision Tree	96.47%
Random Forest	97.29%

Table 6.4: PSO results on merged dataset

6.2.3 Summary of Feature Selection on Merged Dataset

39 Observation: PSO achieved the best trade-off between feature reduction and classification accuracy, making it the most effective method.

Method	Features	Random Forest	Decision Tree	Naïve Bayes
No FS	297	**97.74%**	**96.93%**	**90.78%**
Z-Test	281	97.45%	96.50%	89.02%
MIFS	100	97.11%	95.93%	87.75%
PSO	**55**	**97.29%**	**96.47%**	**88.75%**

Table 6.5: Comparison of feature selection methods on merged dataset

6.3 Detection With Individual Features

While the primary focus is on the merged dataset, a brief comparison with individual and paired datasets is shown below:

Dataset	RF Accuracy
Hardware	66.91%
Permissions	95.48%
Intents	80.35%
Hardware + Permissions	95.4%
Intents + Permissions	94.25%
Intents + Hardware	81.05%

Table 6.6: Accuracy comparison of individual and paired datasets

Conclusion

Initially, the dataset contained 297 features. After applying feature selection techniques, the number of features was reduced to 281 using Z-Test, 100 using MIFS, and 55 using PSO. The accuracy of the models showed slight variations after feature selection. Without feature selection, The Random Forest model attained the peak accuracy of 97.74%. With Z-Test, Decision Tree and Random Forest retained high accuracy at 96.50% and 97.45%, respectively. MIFS improved Decision Tree accuracy to 95.93%, while PSO maintained Random Forest at 97.29%. Feature selection significantly reduced dimensionality while preserving model performance, with PSO achieving the most significant feature reduction (55 features) while keeping high accuracy.

The study demonstrates that well-applied feature selection can enhance model efficiency by reducing computational complexity without sacrificing significant predictive power.

Future Directions and Societal Impact

This study opens up exciting opportunities for additional research in the area of Android malware detection using advanced feature selection techniques. Future research can build upon these findings to develop more efficient and scalable malware detection systems that address the growing complexity of cyber threats. Potential directions include:

- **Innovative Hybrid Feature Selection Approaches:** Researchers can explore novel combinations of statistical and optimization-based feature selection methods to identify even more meaningful features. This could lead to enhanced detection accuracy and faster processing times, making malware detection more practical for real-world applications.
- **Integration with Real-Time and Adaptive Systems:** The evolving nature of malware requires detection systems that can adapt in real time. Future studies may focus on embedding feature selection within dynamic machine learning models that learn and update continuously to keep pace with emerging threats.
- **Expansion to Multi-Source Data Analysis:** Combining binary data with behavioral, network, or system-level features can enrich the detection framework. Future research can investigate multi-modal data integration, which promises to uncover complex malware signatures that single-source analysis might miss.
- **Focus on Explainability and User Trust:** Developing interpretable feature selection models will enable cybersecurity professionals to understand and trust automated detection systems. Future work can emphasize transparent algorithms that offer clear insights into how decisions are made, fostering wider adoption.
- **Early Detection of Threats:** The system helps in identifying malicious apps at an early stage, reducing the risk of data breaches, financial frauds, and personal privacy invasion for users.
- **User-Friendly Security:** It simplifies malware detection for common users, especially those without technical backgrounds, by automating the process through machine learning.
- **Reduced Cybercrime Risk:** By detecting and filtering out harmful applications, the model contributes to reducing broader cybercrimes that rely on malware for illegal access and operations.
- **Lightweight and Efficient:** The model is designed to be computationally light, making it suitable for use in low-end or budget smartphones without compromising performance.

- **Wider Applicability:** It can assist app store platforms in screening applications before making them available to the public, ensuring a safer app ecosystem.
- **Inclusive Protection:** It promotes digital safety for all sections of society, including rural or low-income users who may not have access to expensive cybersecurity tools.
- **Encouraging Safe Digital Practices:** Awareness and use of such technology can help in promoting safer internet usage and increase trust in mobile applications.
- **Foundation for Future Work:** This work creates a base for expanding similar techniques to other smart devices like wearables and IoT systems, which are becoming common in everyday life.

Overall, the continuous advancement in feature selection methods offers a promising avenue for research, with the potential to greatly enhance malware detection systems and promote a more secure digital environment.

Bibliography

[1] H. -H. Kim and M. -J. Choi, "Linux kernel-based feature selection for Android malware detection," The 16th Asia-Pacific Network Operations and Management Symposium, Hsinchu, Taiwan, 2014, pp. 1-4.

[2] R. Adriansyah, P. Sukarno and A. A. Wardana, "Android Malware Detection Using Ensemble Learning and Feature Selection with Insights from SHAP Explainable AI," 2024 11th International Conference on Soft Computing & Machine Intelligence (ISCMI), Melbourne, Australia, 2024, pp. 187-192.

[3] D. J, N. J and N. P, "Multimodal Feature Selection for Android Malware Detection Classifiers," 2022 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), Chennai, India, 2022, pp. 1-5.

[4] T. Eom, H. Kim, S. An, J. S. Park and D. S. Kim, "Android Malware Detection Using Feature Selections and Random Forest," 2018 International Conference on Software Security and Assurance (ICSSA), Seoul, Korea (South), 2018, pp. 55-61.

[5] A. Fatima, R. Maurya, M. K. Dutta, R. Burget and J. Masek, "Android Malware Detection Using Genetic Algorithm based Optimized Feature Selection and Machine Learning," 2019 42nd International Conference on Telecommunications and Signal Processing (TSP), Budapest, Hungary, 2019, pp. 220-223.

[6] S. J. K., S. Chakravarty and R. K. Varma P., "Feature Selection and Evaluation of Permission-based Android Malware Detection," 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), Tirunelveli, India, 2020, pp. 795-799.

[7] S. Khalid and F. B. Hussain, "Evaluating Dynamic Analysis Features for Android Malware Categorization," 2022 International Wireless Communications and Mobile Computing (IWCMC), Dubrovnik, Croatia, 2022, pp. 401-406.

[8] F. Guyton, W. Li, L. Wang and A. Kumar, "Analysis of Feature Selection Techniques for Android Malware Detection," SoutheastCon 2022, Mobile, AL, USA, 2022, pp. 96-103.

[9] M. Nezhadkamali, S. Soltani and S. A. H. Seno, "Android malware detection based on overlapping of static features," 2017 7th International Conference on Computer and Knowledge Engineering (ICCKE), Mashhad, Iran, 2017, pp. 319-325.

[10] J. Wang, B. Li and Y. Zeng, "XGBoost-Based Android Malware Detection," 2017 13th International Conference on Computational Intelligence and Security (CIS), Hong Kong, China, 2017, pp. 268-272.

[11] M. Dhalaria and E. Gandotra, "Android Malware Detection using Chi-Square Feature Selection and Ensemble Learning Method," 2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC), Waknaghat, India, 2020, pp. 36-41.

[12] A. A. Sahal, S. Alam and I. Soğukpinar, "Mining and Detection of Android Malware Based on Permissions," 2018 3rd International Conference on Computer Science and Engineering (UBMK), Sarajevo, Bosnia and Herzegovina, 2018, pp. 264-268.

[13] J. Zhu, Z. Wu, Z. Guan and Z. Chen, "API Sequences Based Malware Detection for Android," 2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops, Beijing, China, 2015, pp. 673-676.

[14] N. Tarar, S. Sharma and C. R. Krishna, "Analysis and Classification of Android Malware using Machine Learning Algorithms," 2018 3rd International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 2018, pp. 738-743.

[15] N. Awasthi, P. R. Gautam and A. K. Sharma, "RFECV-DT: Recursive Feature Selection with Cross Validation using Decision Tree based Android Malware Detection," 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kamand, India, 2024, pp. 1-6.

[16] F. Guyton, W. Li, L. Wang and A. Kumar, "Android Feature Selection based on Permissions, Intents, and API Calls," 2022 IEEE/ACIS 20th International Conference on Software Engineering Research, Management and Applications (SERA), Las Vegas, NV, USA, 2022, pp. 149-154.

[17] M. Nivaashini, R. S. Soundariya, H. Vidhya Shri and P. Thangaraj, "Comparative Analysis of Feature Selection Methods and Machine Learning Algorithms in Permission based Android Malware Detection," 2018 International Conference on Intelligent Computing and Communication for Smart World (I2C2SW), Erode, India, 2018, pp. 72-77.

[18] S. Sharma, Prachi, R. Chhikara and K. Khanna, "Dynamic analysis based Android Malware Detection using ANN and RFE Feature Selection," 2023 5th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2023, pp. 1504-1509.

[19] Z. D. Patel, "Malware Detection in Android Operating System," 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Greater Noida, India, 2018, pp. 366-370, doi: 10.1109/ICACCCN.2018.8748512.

[20] A. Kadir and S. K. Peddoju, "Poster: Android Malware Detection using Hybrid Features and Machine Learning," 2024 IEEE 21st International Conference on Mobile Ad-Hoc and Smart Systems (MASS), Seoul, Korea, Republic of, 2024, pp. 494-495.

[21] X. Li, J. Liu, Y. Huo, R. Zhang and Y. Yao, "An Android malware detection method based on AndroidManifest file," 2016 4th International Conference on Cloud Computing and Intelligence Systems (CCIS), Beijing, China, 2016, pp. 239-243.

[22] A. A. Sahal, S. Alam and I. Soğukpinar, "Mining and Detection of Android Malware Based on Permissions," 2018 3rd International Conference on Computer Science and Engineering (UBMK), Sarajevo, Bosnia and Herzegovina, 2018, pp. 264-268.

[23] R. B. Hadiprakoso, H. Kabetta and I. K. S. Buana, "Hybrid-Based Malware Analysis for Effective and Efficiency Android Malware Detection," 2020 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS), Jakarta, Indonesia, 2020, pp. 8-12.

[24] T. Lu and S. Hou, "A Two-Layered Malware Detection Model Based on Permission for Android," 2018 IEEE International Conference on Computer and Communication Engineering Technology (CCET), Beijing, China, 2018, pp. 239-243.

[25] W. Park, S. -j. Kim and W. Ryu, "Detecting malware with similarity to Android applications," 2015 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea (South), 2015, pp. 1249-1251.

[26] E. Baghirov, "Comprehensive Framework for Malware Detection: Leveraging Ensemble Methods, Feature Selection and Hyperparameter Optimization," 2023 IEEE 17th International Conference on Application of Information and Communication Technologies (AICT), Baku, Azerbaijan, 2023, pp. 1-5.