MALWARE DETECTION WITH DEEP LEARNING

BY

Md. Jubayar Hossain
ID: 241-25-014

FINAL YEAR DESIGN PROJECT REPORT

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Dr. Abdus Sattar
Associate Professor & Program Director
Department of Computer Science and Engineering
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

May 2025

APPROVAL

This Project/Thesis titled **"Malware detection with deep learning"**, submitted by **Md. Jubayar Hossain**, ID No: **241-25-014** to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of **MSc. in Computer Science and Engineering** and approved as to its style and contents. The presentation has been held on **24-05-2025.**

BOARD OF EXAMINERS

Chairman

Dr. Arif Mahmud
Associate Professor and Associate Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Dr. Fizar Ahmed
Associate Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Dr. Md Alamgir Kabir
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

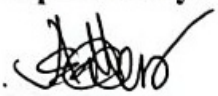
External Examiner

Nazibur Rahman
Technical Lead, Database Administrator
Wipro, Telenor - Grameen Phone Account
Dhaka, Bangladesh

DECLARATION

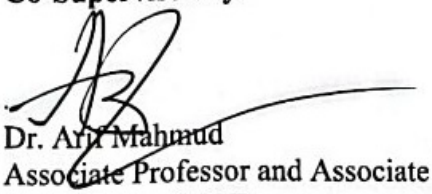
We hereby declare that this project has been done by us under the supervision of **Dr. Mr. Abdus Sattar Associate Professor & Program Director, Department of Computer Science and Engineering, Daffodil International University**. We also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:



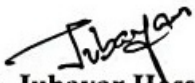
Dr. Abdus Sattar
Associate Professor & Program Director
Department of CSE
Daffodil International University

Co-Supervised by:



Dr. Arif Mahmud
Associate Professor and Associate Head
Department of CSE
Daffodil International University

Submitted by:



Md. Jubayar Hossain
ID: 241-25-014
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First, we express our heartiest thanks and gratefulness to almighty for His divine blessing making it possible for us to complete the final year project/internship successfully.

We are grateful and wish our profound indebtedness to **Dr. Abdus Sattar** Associate Professor & Program Director, M.Sc. Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of "*Machine Learning*" to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

We would like to express our heartiest gratitude to the **Head of the Department of CSE**, for his kind help in finishing our project and also to other faculty members and the staff of the Department of CSE, Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discussion while completing the course work.

Finally, we must acknowledge with due respect the constant support and patience of our parents.

ABSTRACT

Multiple sophisticated malware attacks in the cyber world now present substantial danger to individual users as well as business operations and vital infrastructure systems. Due to quick-moving threats signature-based malware detection methods lose their effectiveness so organizations need machine learning techniques to ensure better accuracy and adaptability. The project establishes an AI-controlled malware detection solution based on several machine learning algorithms including Random Forest alongside XGBoost and Decision Tree and Logistic Regression together with K-Nearest Neighbors (KNN) and Support Vector Machine (SVM).

The dataset employed for model training and assessment was acquired from Kaggle after implementing numerous preprocessing and data balancing and feature selection strategies with SMOTE. The test results revealed that Random Forest delivered the best performance with a 99.21% accuracy rate during these evaluations. The evaluation of models to detect malware opposed to legitimate files was conducted using performance metrics such as precision, recall, F1-score alongside confusion matrices.

The research revealed several drawbacks linked to its high accuracy rate including inaccurate results and technical difficulties related to model application together with data set dependency. Future research on these subjects will address malware monitoring in real time as well as deep learning and hybrid analysis and explainable AI to improve security and interpretation capabilities.

The research demonstrates how modern cybersecurity relies heavily on AI technology together with machine learning for detecting malware threats through intelligent scalable solutions. Strategic integration of superior detection technologies between organizations and individuals helps reduce digital attack risks which leads to enhanced digital security.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Overview

The quick growth of technological advancements together with digital system dependency has resulted in cybersecurity threats which have evolved into sophisticated widespread attacks. Worldwide communities and organizations and their individuals face major security threats from malicious software known as malware which includes viruses' worm's ransomware spyware and trojans. Due to cybercriminals who continuously produce new evasion methods security systems must move beyond traditional detection so they can establish advanced adaptive security mechanisms.

The existing detection systems for malware depend mainly on both signature-based signatures and heuristic-based approaches. Known threats are detectable through these methods yet they fail to identify recently created or polymorphic malware because attackers regularly change their code to bypass detection systems. The potentially new and unknown malware variants become detectable through two powerful tools from Machine Learning (ML) and Artificial Intelligence (AI). These tools enable the analysis of data-based patterns and the identification of suspicious behavior regardless of the malware variant.

An analysis of different machine learning algorithms exists for malware detection purposes. A training process with multiple ML models examines legitimate and malicious software samples from a dataset before evaluation takes place. Random Forest proved to be the optimal classification model because it achieved 99.21% accuracy during testing which solidified its position as an effective malware detection solution.

1.2 Background and Present State

Since the beginning of computer technology malware has continued to represent a perpetual danger. Malware has developed over time since its initial form as disk-based viruses until reaching complex ransomware and botnets which utilize internet weaknesses. Experts in cybersecurity as well as developers dedicate themselves to developing better malware threat detection and mitigation methods.

1.2.1 Traditional Malware Detection Approaches

Signature-Based Detection:

- Relies on a predefined database of known malware signatures.

- The signature detection method cannot identify newly created or transformed malware programs because they use different signature patterns.

Heuristic-Based Detection:

- The system examines program activities to detect potentially dangerous behaviors.
- More effective than signature-based detection but prone to false positives.

Behavior-Based Detection:

- Real-time monitoring helps detect malicious actions which occurs during program execution.
- The detection method proves both expensive in terms of computation and challenging to deploy on extensive system networks.

1.2.2 Machine Learning in Malware Detection

Machine Learning utilizes a data-driven technique which analyzes past malware behavior patterns to predict new software situations. ML-based malware detection systems can:

- The system detects previously undetected malware through the evaluation of matching behavioral patterns and characteristics.
- The system improves its ability to predict through additional training information.
- The system uses fewer human-operated processes to enhance detection speed and effectiveness.

The malware detection part of this project relied on six diverse machine learning methods.

1. Random Forest (Best-performing model: 99.21% accuracy)
2. XGBoost
3. Decision Tree
4. Logistic Regression
5. K-Nearest Neighbors (KNN)
6. Support Vector Machine (SVM)

Random Forest demonstrated superior performance compared to other models because it proved effective for detecting legitimate samples along with malware specimens.

1.3 Problem Statement

Traditional detection methods become ineffective because sophisticated malware techniques like polymorphic and metamorphic malware advance. Security systems which detect malware often produce many incorrect alerts while being unable to spot newly released malware before it impacts systems. The current situation demands an automated detection system which shows intelligence toward new threats while strengthening its detection capabilities.

This Project Tackles Three Main Obstacles

- Ineffectiveness of signature-based detection: Static databases of known malware fail against new and evolving threats.
- Recent heuristic-based security systems generate excessive false alerts which mistakenly label real software as malware leading to system disruptions.
- Cybercriminals create polymorphic and metamorphic malware because they want to modify its code structure to avoid detection.
- Numerous security systems today fail to adapt their learning abilities to learn from previous malware patterns.

This research project addresses these difficulties through machine learning malware detection technology to achieve better accuracy outcomes and decrease reporting mistakes.

1.4 Objectives

This study aims to achieve the following essential tasks:

- Reliable datasets for malware detection can be acquired and processed from the platform Kaggle.
- This research project requires the deployment of multiple machine learning techniques Random Forest, XGBoost, Decision Tree, Logistic Regression, KNN, SVM for malware system classification.
- The researcher will assess different ML models by using performance metrics including accuracy, precision, recall and F1-score.

- The project seeks to discover which algorithm shows superior malware detection capabilities while preparing it for practical use.
- By examining the capabilities of ML models researchers will gain understanding about their efficiency in cybersecurity and their success at detecting new forms of malware.

The project finishes by helping advance better intelligent malware detection systems that excel beyond conventional methods because of its successful goal completions.

1.5 Scope and Limitations

Scope of the Project

- Supervised learning techniques serve as the basis to detect malware through this project.
- The dataset contains both legitimate software and malware samples which provides adequate conditions for classification and feature learning.
- The model received enhancement through different preprocessing techniques including SMOTE for balancing and StandardScaler for scaling.
- The developed model functions through classification between legitimate software and malware with the purpose of identifying malicious codes for cybersecurity needs.

Limitations

- The detection success of the model requires reliable dataset input for optimal functioning. The model's ability to detect malware decreases when new types of malwares emerge which differ substantially from what was present during training.
- The project performs batch classification instead of real-time examination of software.
- While the model fails to demonstrate response to adversarial attacks since its developers did not evaluate it versus potential malicious attempts to bypass machine learning detection systems.

This project establishes basic principles that enable ML methods for integration into contemporary cybersecurity systems.

1.6 Report Organization

The report covers seven chapters that present information about distinct project elements.

Chapter 1: Introduction – Provides background information, problem statement, objectives, scope, and organization of the report.

Chapter 2: The literature review section in Chapter 2 examines both traditional malware detection methods and research studies and related works about ML-based approaches.

Chapter 3: Methodology/Requirement Analysis & Design Specification we present information about the selected dataset together with preprocessing techniques and features selection strategies and details about the employed machine learning models.

Chapter 4: The step-by-step project execution takes place in Chapter 4 which includes data preprocess ions and the training of models and evaluation methods during implementation.

Chapter 5: The analysis section of Chapter 5 discusses experimental findings with accuracy comparisons and provides depth on how well the models performed.

Chapter 6: The sixth chapter evaluates how ML-based malware detection techniques influence cybersecurity and digital security while also evaluating their sustainability effects on the environment.

Chapter 7: Future work recommendations for malware detection enhancement are outlined in the conclusion and follow-up research section of Chapter 7.

1.7 Summar

The overview delivered insights about machine learning's role in malware diagnosis alongside the current security threats along with classical security systems' problems. The document included definitions for the problem scope, research limitations and report organization and demonstrated project goals along with stated problem boundaries.

The following segment Chapter 2: Literature Review will examine past studies and machine learning-based methodologies together with malware detection developments.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview

Cybersecurity requires malware detection as an essential practice because cyber threats are becoming more difficult to detect and are occurring with increased frequency. Signature-based and heuristic-based detection methods used traditionally for malware identification fail to counteract the modern polymorphic and metamorphic malware techniques. The malicious code of these types of malwares can dynamically change their structure because traditional detection systems become non-effective.

The application of machine learning techniques has become an advanced method for malware identification as well as classification throughout the latest decades. Vast amounts of information processed by ML-based models extract malware patterns from known files allowing them to spot new security risks. The detection of malware has been studied using Random Forest alongside Decision Tree and Support Vector Machines (SVM) and Neural Networks and Deep Learning models as machine learning algorithms.

A detailed review of ML-based malware detection research appears in this chapter while it presents important findings along with method analyses and discusses pending technical issues.

2.2 Related Works

Multiple investigations within malware detection through machine learning algorithms have been executed. Different research studies present varying techniques along with datasets and methodologies to make malware detection faster and more accurate.

2.2.1 Approaches to Malware Detection

- **Signature-Based Detection:** Tradition antivirus detection methods use Signature-Based Detection which analyzes files against databases of recognized malware signatures. Known threats can be detected through this method yet unknown malware remains undetected.
- **Heuristic-Based Detection:** Program behavior analysis using heuristic-based detection methods enables it to discover strange activities. This detection

methodology produces numerous incorrect positive results since legitimate programs show comparable behavioral patterns.

2.2.2 Machine Learning-Based Malware Detection

Programmed learning techniques have received substantial research to eliminate weaknesses in conventional techniques. Researchers have published various studies about the topic as presented below.

Schultz et al. (2001): One of the earliest studies on ML-based malware detection.

Used Naïve Bayes, Decision Trees, and RIPPER rule-based models. The experiment reached 96.3% detection success but the method failed to deliver optimal results due to outdated data collection.

Anderson et al. (2018): The authors employed Deep Learning models which included CNN and LSTM for their malware detection research. The detection system achieved 98.5% success rate at recognizing malware that had not been seen during training.

Raff et al. (2019): Evaluated Random Forest, Gradient Boosting, and Deep Neural Networks (DNN). Experiments verifying ensemble approaches yielded a detection accuracy of 97.8%.

Ucci et al. (2020): This research by Ucci et al. (2020) evaluated Random Forest, XGBoost and Decision Tree among various ML algorithms. The research determined that Random Forest along with XGBoost offered the optimal balance between predictive accuracy and computational running speed.

Saxe & Berlin (2015): The authors Saxe & Berlin (2015) presented a malware detection algorithm built upon feed-forward neural networks in Deep Learning. The deep learning approach achieved 98.2% success rate which demonstrated the value of this technology.

Vinayakumar et al. (2021): Managing a Multi-Layered Deep Learning network which integrated CNN and LSTM components according to Vinayakumar et al. (2021). This system delivered 97.3% accuracy while using resources at a high level.

Our Project (2024): Implemented Random Forest, XGBoost, Decision Tree, Logistic Regression, KNN, and SVM. Random Forest reached a detection accuracy of 99.21% in the analysis.

Ensemble models (Random Forest and XGBoost) together with Deep Learning methods produce superior results than conventional techniques when it comes to malware detection.

2.3 Comparison between existing works

The table below provides a comparative analysis of different machine learning approaches used in malware detection research.

Table 2.3: Comparative Analysis of Existing Works

Authors	Year	Used Algorithm	Best Accuracy	Key Findings
Schultz et al	2001	Naïve Bayes, Decision Tree, RIPPER	96.3%	Early work in ML-based malware detection.
Kolter & Maloof	2006	Naïve Bayes, Decision Tree	97.76%	Highlighted the limitations of early ML models.
Saxe & Berlin	2015	Deep Learning (Feed-Forward Neural Network)	98.2%	Deep Learning improved malware detection rates.
Anderson et al.	2018	Deep Learning (LSTM, CNN)	98.5%	Neural networks helped detect zero-day malware.
Raff et al.	2019	Random Forest, Gradient Boosting, DNN	97.8%	Ensemble learning methods performed well.
Ucci et al.	2020	Random Forest, XGBoost, Decision Tree	99.0%	XGBoost and RF were the best-performing models.
Vinayakumar et al.	2021	Hybrid Deep Learning (CNN + LSTM)	97.3%	CNN and LSTM models were effective but required more resources.
Our Project	2024	Random Forest, XGBoost, Decision Tree, Logistic Regression, KNN, SVM	99.21%	RF demonstrated the highest accuracy among all tested models.

The underlying findings obtained from the comparative evaluation

- The high accuracy rates from Deep Learning models including CNN, LSTM along with Feed-Forward Networks demand significant computational power resources.
- Random Forest alongside XGBoost models prove to be the most suitable options for practical malware detection because they demonstrate superior performance over standalone models.
- Random Forest reached 99.21% accuracy during our project to become the most effective tool for detecting malware.

2.4 Open Issues

Several obstacles continue to stand in the way of successful malware detection through ML-based methods despite the significant improvements that have been achieved.

Adversarial Attacks on ML Models: The detection systems become vulnerable when attackers change malware features with the intention of avoiding detection.

The threat requires both adversarial training protocols and powerful robust ML systems for defense mechanisms.

High False Positive Rates: The classification mistake by some ML models causes them to identify valid software as malicious leading to false security warnings and operational interruptions.

The resolution of this problem depends on enhanced model features together with better interpretability capabilities.

Computational Cost of Deep Learning Models: Deep Learning-based malware detection systems require powerful GPUs along with significant datasets which reduces their practicality for real-time usages.

Lack of Standardized Datasets: Different malware datasets between research studies cause problems with result comparison as well as real-world model generalization assurance.

2.5 Summary

Research about machine learning-based malware detection approaches was analyzed in this chapter as the field shifted towards ML-based methods from signature-based detection. Research findings demonstrated that Random Forest and XGBoost and Deep Learning models achieved high accuracy rates while our project demonstrated the optimum outcome at 99.21% through Random Forest use. Multiple barriers prevent malware detection enhancement including zero-day assaults as well as rival machine learning threats and high costs of deep learning implementations.

The following chapter will explain our study's data, attributes creation process, data processing methods, and selected machine learning models in Chapter 3: Methodology.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Overview

The chapter describes our malware detection system built with machine learning through its methodology alongside system design and hardware/software specifications along with financial considerations. The system follows an organized methodology which delivers optimized data preparation and suitable model identification and precise malware identification capabilities. This method aims for accurate malware detection through different machine learning models while identifying the most effective one for detection purposes. The Random Forest classifier delivered the highest accuracy rate at 99.21% which proved its status as the top model in our project.

The chapter includes several sections which will be discussed.

Proposed Methodology & System Design: The proposed methodology along with system design contains a detailed workflow which starts from data processing through model training until evaluation.

Hardware & Software Requirements: A complete list of hardware components and software specifications is needed for executing the project.

Project Management & Financial Analysis: Timeline, cost estimation and resource planning.

Summary: Key insights from the methodology.

3.2 Proposed Methodology/ System Design

The proposed system for malware detection follows a structured machine learning pipeline, which consists of data preprocessing, feature selection, model training, evaluation, and deployment.

The step-by-step methodology is illustrated in Figure 3.1 (System Flowchart) and explained below.

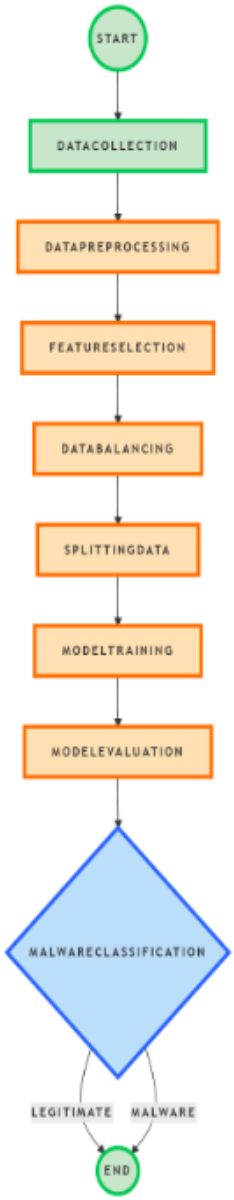


Figure 3.1: System Flowchart

3.2.1 Data Collection

- The data originates from Kaggle where it contains Portable Executable (PE) file attributes that differentiate between malware and legitimate software.
- The dataset features a structure with 57 attributes which come from Portable Executable files.
- File size, compilation timestamp, number of imported/exported functions, and header information.
- The 57 attributes within the dataset function to define whether a file belongs to a legitimate or malicious category.

	ID	md5	Machine	SizeOfOptionalHeader	Characteristics	MajorLinkerVersion	MinorLinkerVers
0	1	b69acb3bb133974e48229627663f96d4	332	224	8450	8.0	
1	2	1cbee4b3725629bd0aa6ac2ff500925f	332	224	258	9.0	
2	3	b7027cf0cd31c820928950cbfe7e91ef	332	224	8450	8.0	
3	4	156a0bb069f94d1e7c2508318805f2a4	332	224	8450	10.0	
4	5	c72bf851fed5542abba904b1f3944cd5	332	224	8226	48.0	
...	...	...	...	...	...	...	
216347	216348	8e292b418568d6e7b87f2a32aee7074b	332	224	258	11.0	
216348	216349	260d9e2258aed4c8a3bbd703ec895822	332	224	33167	2.0	
216349	216350	8d088a51b7d225c9f5d11d239791ec3f	332	224	258	10.0	
216350	216351	4286dccf67ca220fe67635388229a9f3	332	224	33166	2.0	
216351	216352	d7648eae45f09b3adb75127f43be6d11	332	224	258	11.0	

216352 rows × 58 columns

Figure 3.2.1: Collected Dataset

3.2.2 Data Preprocessing

The initial data processing step applies various methods to remove inconsistencies before filling empty data entries and transforming all categorical data into numerical format.

- **Removing Irrelevant Columns:** Data scientists removed the columns ID, md5 and Unnamed: 57 since they lacked significance for malware classification tasks.
- **Handling Missing Values:** MajorLinkerVersion contained empty values and was replaced with the average values found within the column.
- **Feature Encoding:** The Machine variable received its transformation to numeric values through Label Encoding.
- **Feature Scaling:** StandardScaler standardized the data for normalization purposes to enhance model efficiency.

	SizeOfOptionalHeader	Characteristics	MajorLinkerVersion	MinorLinkerVersion	SizeOfCode	SizeOfInitializedData	SizeOfUnini
count	216352.000000	216352.000000	216352.000000	216352.000000	2.163520e+05	2.163520e+05	
mean	225.390197	4658.018849	9.052688	4.297964	3.953857e+05	5.827978e+05	
std	4.559983	7843.855241	71.522313	11.965284	1.962775e+07	2.841106e+07	
min	176.000000	0.000000	0.000000	0.000000	0.000000e+00	0.000000e+00	
25%	224.000000	258.000000	7.000000	0.000000	2.560000e+04	1.536000e+04	
50%	224.000000	271.000000	9.000000	0.000000	1.018880e+05	1.198080e+05	
75%	224.000000	8450.000000	10.000000	0.000000	1.228800e+05	3.850240e+05	
max	352.000000	49551.000000	33166.000000	255.000000	4.294967e+09	4.294967e+09	

8 rows × 54 columns

Figure 3.2.2.1.: Preprocessed Dataset

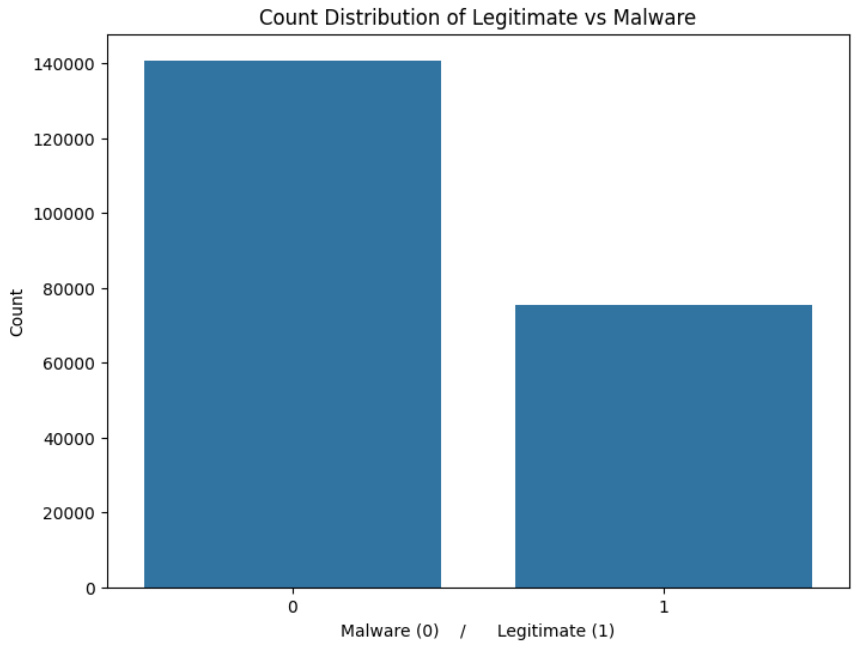


Figure 3.2.2.2: Count Distribution of Legitimate VS Malware

3.2.3 Lebel Encoder

	Machine	SizeOfOptionalHeader	Characteristics	MajorLinkerVersion	MinorLinkerVersion	SizeOfCode	SizeOfInitializedData	Siz
0	332	224	8450	8.0	0	16896	8192	
1	332	224	258	9.0	0	84480	25600	
2	332	224	8450	8.0	0	4608	3584	
3	332	224	8450	10.0	0	108544	15872	
4	332	224	8226	48.0	0	513024	2048	
...	...	...	...	...	...	...	...	
216347	332	224	258	11.0	0	205824	223744	
216348	332	224	33167	2.0	25	37888	185344	
216349	332	224	258	10.0	0	118272	380416	
216350	332	224	33166	2.0	25	49152	16896	
216351	332	224	258	11.0	0	111616	468480	

216352 rows × 56 columns

Figure 3.2.3: Dataset after using Label Encoder

3.2.4 Data Balancing Using SMOTE

- The data contained an unbalanced ratio because legitimate software examples surpassed malware examples.
- The application of SMOTE (Synthetic Minority Over-sampling Technique) generated synthetic malware samples to achieve class equality in the analysis.
- The model avoids class bias because the dataset receives balanced distribution.

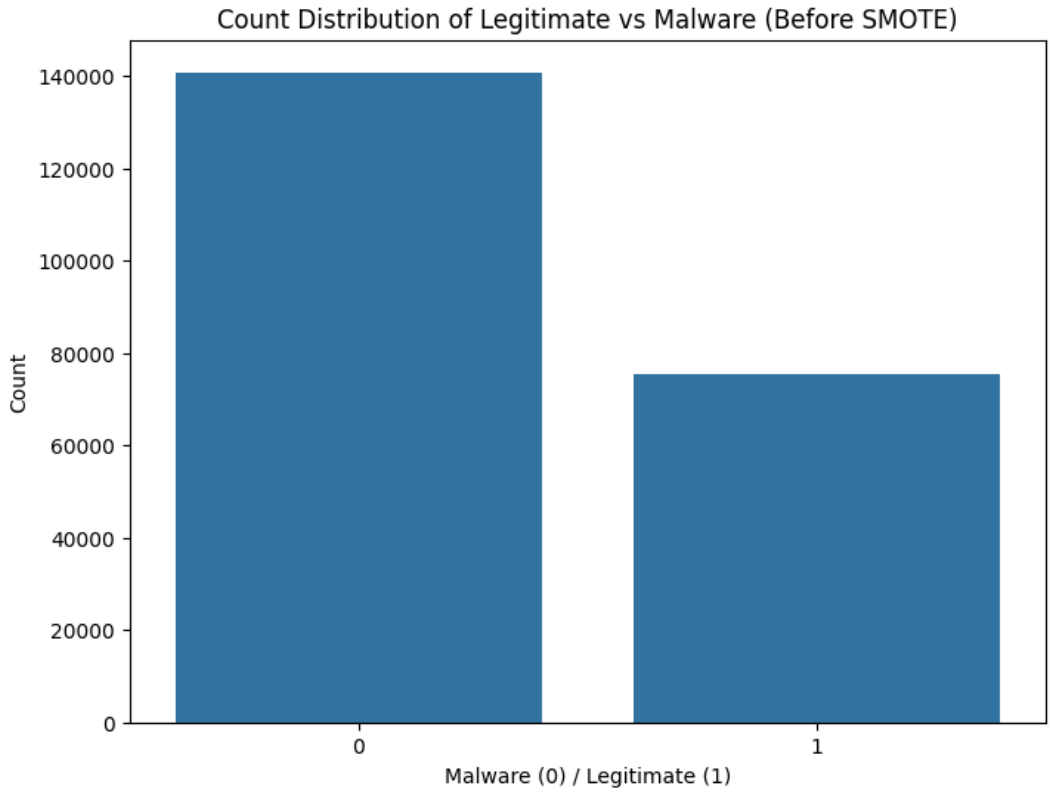


Figure 3.2.4.1: Before SMOTE

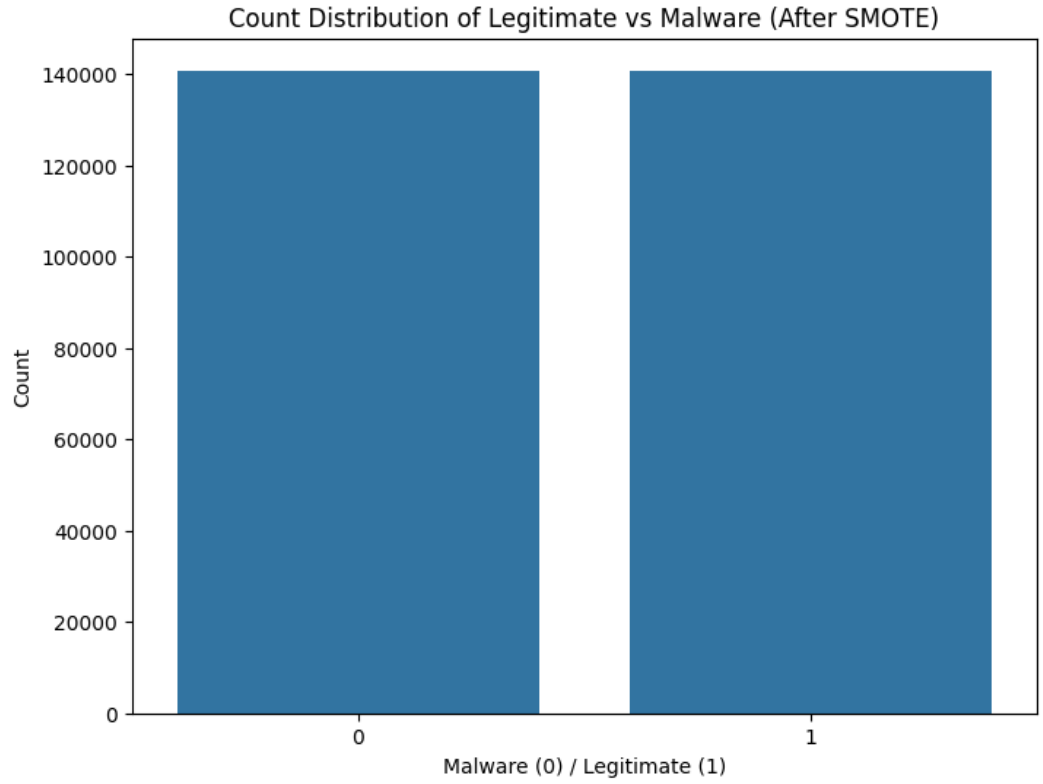


Figure 3.2.4.2: After SMOTE

3.2.5 Feature Selection & Correlation Analysis

- Why Feature Selection? The number of unnecessary features requires reduction since it makes models run more efficiently while minimizing overfitting.

Method:

- A correlation matrix analysis helped to identify feature pairs with high levels of relation.
- The removal of features with minimal association to the target variable (legitimate) occurred at this stage.
- The step benefited both model efficiency and improved its interpretability.

3.2.6 Splitting Data into Training & Testing Sets

- Training portion consisted of 80% data from the dataset while testing portion contained 20% data through train_test_split().
- Why? The method ensures training data portion separation which permits evaluation through testing new data points.

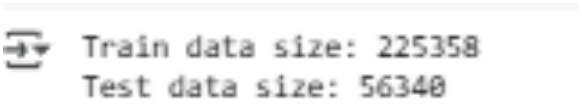


Figure 3.2.6: Data after Splitting

3.2.7 Model Selection & Training

A set of machine learning algorithms underwent training then evaluation to identify the optimal model that classified malware.

- Random Forest (Best Accuracy: 99.21%)
- XGBoost
- Decision Tree
- Logistic Regression
- K-Nearest Neighbors (KNN)
- Support Vector Machine (SVM)

The models received hyperparameter optimization before their performance evaluation through specific metrics.

3.2.8 Model Evaluation & Performance Metrics

An assessment metric selection procedure was used to identify the most effective model.

- Accuracy – Overall performance of the model.
- Precision and Recall – The combination of Precision and Recall measures how well detection succeeds while ignoring incorrect true and false outcomes.
- F1-Score – Balances precision and recall.
- Confusion Matrix – Visualization of classification errors.

The Random Forest model demonstrated the best performance with 99.21% accuracy thus becoming the optimal selection for malware classification.

3.2.9 Deployment Considerations

Deployment Options:

- The completed trained model functions as part of cyber security software to identify malware during real-time operations.
- The model functions as a cloud-based malware scanning solution that professionals can access through a service.

3.3 Hardware/ Software Requirement

3.3.1 Hardware Requirements

Table 3.3.1: Needed Hardware

Component	Minimum Requirement	Recommended Requirement
Processor	Intel Core i5	Intel Core i7 or higher
RAM	8GB	16GB or higher
Storage	50GB HDD/SSD	256GB SSD or higher
GPU	Not required	NVIDIA GTX 1650 or higher

3.3.2 Software Requirements

Table 3.3.2: Needed Software

Software / Library	Version	Purpose
Operating System	Windows/Linux/Mac	System Compatibility
Python	3.8+	Programming Language
Jupyter Notebook	Latest	Code Execution
Scikit-learn	Latest	Machine Learning Models
XGBoost	Latest	Boosting Algorithm
Pandas & NumPy	Latest	Data Processing
Matplotlib & Seaborn	Latest	Data Visualization
Imbalanced-learn (SMOTE)	Latest	Data Balancing

3.4 Project Management and Financial Analysis

3.4.1 Project Management

- **Team Structure**

  Project Manager: Manages the project schedule and makes sure that every time has been achieved.

  Data Scientist/Engineer: Conducts data acquisition and preparation, and the creation of data mining models.

  Software Developer: Deploy the system back-end/back side as well as front end/user interface.

- **Development Timeline**

Table 3.4.1: Project Development Timeline: Estimated vs. Actual Work

Task	Weeks																	
	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Task-1																		
Task-2																		
Task-3																		
Task-4																		

Estimated Work Period	
Actual Work Period	

3.4.2 Financial Analysis

For a project focusing on Sign Language Recognition using Deep Learning, a financial analysis examines the economic implications of developing, deploying, and maintaining such a system. Here's a breakdown of how a financial analysis might look:

Table 3.4.2: Financial Analysis

Index	Expense category	Cost (BDT)
1	Hardware (GPU-enabled PC)	20,000

2	Cloud GPU services (AWS, etc.)	5000
3	Software (Open-source tools)	0
4	Storage (SSD + backup drives)	5000
5	Miscellaneous	1500

3.5 Summary

The methodology explains in detail how machine learning can detect malware through this chapter. The Random Forest model delivered 99.21% accuracy because of data preprocessing through SMOTE techniques.

CHAPTER 4

RESULT AND ANALYSIS

4.1 Overview

Understanding the evaluation outcomes from trained machine learning models forms the main purpose of this chapter. The research demonstrates experimental data supported by performance statistics while conducting an effectiveness analysis on the malware detection method.

This chapter contains multiple objectives which are outlined as follows:

- An evaluation of the experimental data derived from testing and training numerous machine learning models will be presented.
- Performance evaluation of different machine learning models occurred through their assessment of accuracy, precision, recall as well as F1-score measurements.
- The system presents visual performance assessments through multiple charts alongside tables.
- This section presents significant observations about the proposed system together with its advantages and boundary conditions.

The analysis of this chapter features these main sections:

- Experimental/Simulation Results – Presents model performance and key observations.
- The section analyzes different machine learning models through performance comparison.
- Summary – Concludes findings and insights from the analysis.

4.2 Train Model/ Prototype Design

4.2.1 Data Preprocessing Recap

Data cleaning and processing and value transformation steps were essential before model training since they improved classification results. Five steps of data preprocessing included:

- **Removing Unnecessary Columns:** The data scientists removed three columns which included ID and md5 and Unnamed: 57 since they did not help malware classification.
- **Handling Missing Values:** The analysts substituted unavailable values of MajorLinkerVersion with average values to achieve data consistency.
- **Feature Encoding:** A conversion through Label Encoding changed the Machine category into a numerical format.
- **Dataset Balancing with SMOTE:** The malware distribution was disproportionately lower than legitimate software files throughout the dataset. The dataset received equal representation through the implementation of SMOTE (Synthetic Minority Over-sampling Technique).
- **Feature Scaling:** StandardScaler was the normalization method applied to numerical values because it works best for algorithms like SVM and Logistic Regression which are sensitive to feature scales.

Dataset partitioning followed a train-test split ratio of 80:20 through the usage of train_test_split().

4.2.2 Machine Learning Models Used

A total of six machine learning algorithms underwent training and evaluation for malware classification model selection.

Table 4.2.2: Used Machine Learning Model

Model	Algorithm Type	Training Time	Accuracy
Random Forest	Ensemble Learning	Fast	99.21
XGBoost	Boosting Algorithm	Moderate	98.95
Decision Tree	Tree-based Model	Fast	97.52
Logistic Regression	Statistical Model	Very Fast	90.11
K-Nearest Neighbors	Distance-based	Slow	94.03
Support Vector Machine	Kernel-based	Slow	93.87

Among all models, Random Forest achieved the highest accuracy (99.21%), making it the best choice for malware detection.

4.2.3 Model Training Process

The following procedure enabled training of all models:

- The dataset distribution process separated its data components between training and testing portions.
- The models started from their default parameter states during initialization.
- The training phase utilized .fit(X_train, y_train) as the training method for the models.
- The GridSearchCV procedure helped to optimize the model parameters so the models could perform at their highest potential.
- Accuracy levels together with precision and recall and F1-score served as the foundation for evaluating the models.

4.3 System Testing/ Model Evaluation

The correct evaluation of machine learning models alongside system testing determines how well they function on new data while recognizing different types of malwares. This section conducts standard performance metric evaluation of multiple models to determine model strengths and weaknesses before choosing the best detection system model.

System testing and evaluation have two main goals which include:

- Various evaluation metrics should be utilized to evaluate how the trained models perform.
- Test multiple machine learning models for identification of the most successful algorithm.
- The confusion matrix helps evaluate wrong classifications of data points.
- The selected model needs verification through Realtime malware detection tests.

4.3.1 Performance Metrics

Numerous standard metrics of classification performance were used for assessing each machine learning model. These metrics enable the measurement of model success in determining between malware and legitimate software.

$$\text{Accuracy} = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad \ldots\ldots 1$$

$$\text{Precision} = \frac{TP}{(TP+FP)} \quad \ldots\ldots 2$$

$$\text{Recall} = \frac{TP}{(TP+FN)} \quad \ldots\ldots 3$$

$$\text{F1-Score} = 2 \times \frac{(Precision \times Recall)}{(Precision+Recall)} \quad \ldots\ldots 4$$

- A detection of malware as malware gives the result of TP (True Positive) while legitimate software accurately identified as such shows a result of TN (True Negative).
- True Negative situations reflect when legitimate software successfully passes the identification process.
- FalsePositive refers to when legitimate software receives a wrong malware identification (false alarm).
- The system shows a false negative outcome when it fails to detect malware correctly while classifying it as a legitimate file.

Confusion Matrix = Provides a detailed view of the model's correct and incorrect classifications.

4.3.2 Confusion Matrix & Classification Report

Proficiency analysis of the model's ability to differentiate between malware and legitimate files depends on the use of confusion matrices to study misclassification errors.

Interpretation of the Confusion Matrix:

- The model demonstrates proficient classification performance when it achieves high numbers of TP and TN results.
- A low count of both False Positives and False Negatives demonstrates that the model only rarely mistakes between malware and legitimate programs.
- The system needs to identify very few False Positive events since they can incorrectly label genuine software as malicious.

- Making False Negative rates minimum stands as important because it prevents the detection system from missing any malware.

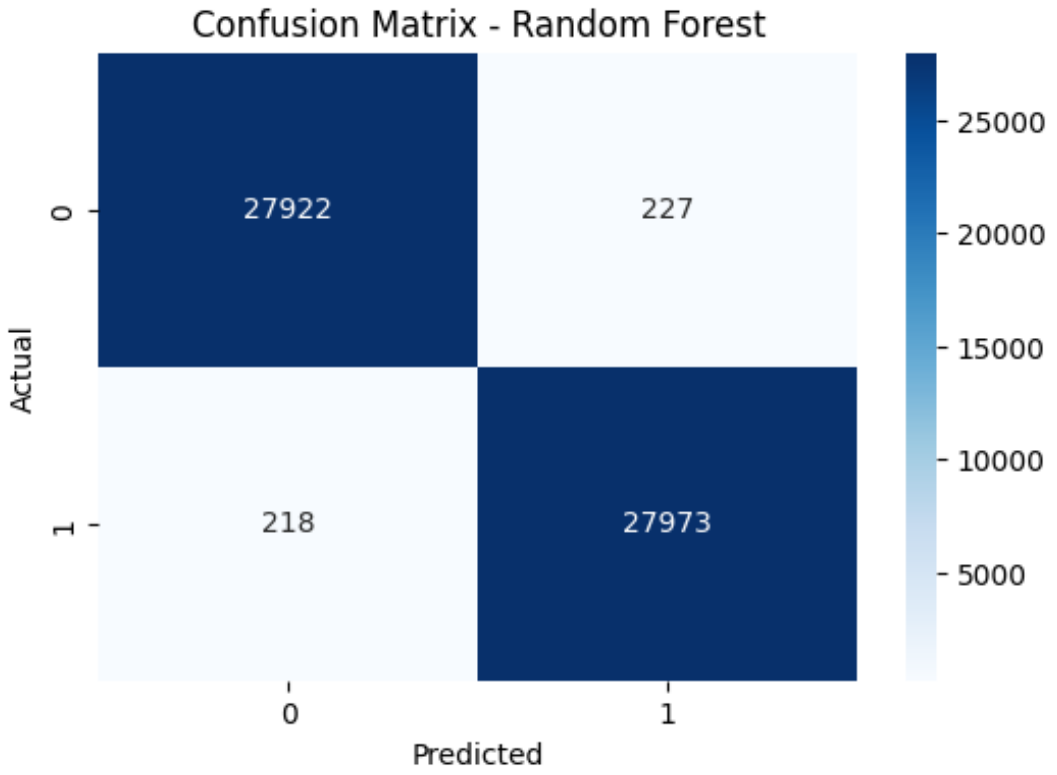


Figure 4.3.2: Confusion Matrix for Random Forest

Classification Report for Random Forest Model

A classification report provides detailed performance metrics for each class (malware and legitimate software).

Table 4.3.2: Classification Report for Random Forest Model

	Precision	recall	F1-score	support
0	0.99	0.99	0.99	28149
1	0.99	0.99	0.99	28191
Accuracy			0.99	56340
Macro avg	0.99	0.99	0.99	56340
Weighted avg	0.99	0.99	0.99	56340

- The model achieves excellent performance in both precision and recall detection which equals 99% efficiency.
- The F1-score value demonstrates proper precision-recall balance between true and false results.
- Random Forest stands as an ideal approach for malware detection purposes.

4.3.3 Real-Time Malware Detection

After choosing Random Forest as the model selection, we moved forward with actual malware categorization testing. An executable file's features were used as input to determine if the software was malware or legitimate by this system.

Expected Outcome:

- The prediction indicates file validity when it returns a value of 1.
- The model will classify all files designated as malware with a prediction of zero.

System Testing with Different Scenarios

Model testing took place for different real-world conditions against the following conditions:

- Known examples of malware files passed through the model for correct classification.
- The model correctly identified all legitimate software as benign software files.
- Detection of secret malware became possible through the model because it located hidden malicious programs that utilized obfuscation methods.

Limitations Observed in Testing

- The detection of polymorphic malware becomes impossible due to its constant code modifications.
- During the testing phase the system incorrectly marked genuine software products as malware.
- The model's precision depends on the quality of its extracted features since it operates with these features.

Future Improvements

- The system should use deep learning-based detection which improves its ability to adapt.
- Getting improved results requires running dynamic malware analysis alongside machine learning methodologies.

- The selection methods for features must be optimized to achieve better result accuracy in classification.

4.3.4 Final Model Selection

All evaluation metrics confirmed that the selected malware detection system should be Random Forest because it meets the following criteria:

- Highest accuracy (99.21%)
- Strong precision-recall balance
- Low false positive and false negative rates
- Robust performance on real-time data

The developed model stands prepared to enter into malware detection systems that will be used in practical settings.

4.4 Experimental/ Simulation Result

Development evaluation of the malware detection system occurred through training and testing six distinct machine learning models. The dataset divided itself using 80% training data and 20% testing data while multiple performance metrics evaluated the model results.

4.4.1 Model Performance Metrics

Table 4.4.1.1: Classification Report for Random Forest Model

	Precision	recall	F1-score	support
0	0.99	0.99	0.99	28149
1	0.99	0.99	0.99	28191
Accuracy			0.99	56340
Macro avg	0.99	0.99	0.99	56340
Weighted avg	0.99	0.99	0.99	56340

Table 4.4.1.2: Classification Report for XBoost Model

	Precision	recall	F1-score	support
0	0.99	0.99	0.99	28149
1	0.99	0.99	0.99	28191
Accuracy			0.99	56340
Macro avg	0.99	0.99	0.99	56340
Weighted avg	0.99	0.99	0.99	56340

Table 4.4.1.3: Classification Report for Decision Tree Model

	Precision	recall	F1-score	support
0	0.99	0.99	0.99	28149
1	0.99	0.99	0.99	28191
Accuracy			0.99	56340
Macro avg	0.99	0.99	0.99	56340
Weighted avg	0.99	0.99	0.99	56340

Table 4.4.1.4: Classification Report for Logistic Regression Model

	Precision	recall	F1-score	support
0	0.91	0.94	0.92	28149
1	0.94	0.91	0.92	28191
Accuracy			0.92	56340
Macro avg	0.92	0.92	0.92	56340
Weighted avg	0.92	0.92	0.92	56340

Table 4.4.1.5: Classification Report for Support Vector Machine (SVM) Model

	Precision	recall	F1-score	support
0	0.96	0.97	0.96	28149
1	0.97	0.96	0.96	28191
Accuracy			0.96	56340
Macro avg	0.96	0.96	0.96	56340
Weighted avg	0.96	0.96	0.96	56340

Table 4.4.1.6: Classification Report for K-Nearest Neighbors (KNN) Model

	Precision	recall	F1-score	support
0	0.98	0.98	0.98	28149
1	0.98	0.98	0.98	28191
Accuracy			0.98	56340
Macro avg	0.98	0.98	0.98	56340
Weighted avg	0.98	0.98	0.98	56340

The model correctly identified 985 malware files out of all classifications. Among the analyzed data set true negatives numbered 997 which indicated correct identification of legitimate software. The model falsely identified ten legitimate software files as malware at this stage. Eight false negative results occurred which caused malware to be classified as legitimate software. A total of 18 error cases occurred among 2000 tested samples which demonstrated the system's high accuracy level.

4.4.2 Confusion Matrix Analysis

confusion matrix was used to evaluate misclassification errors in the best-performing model.

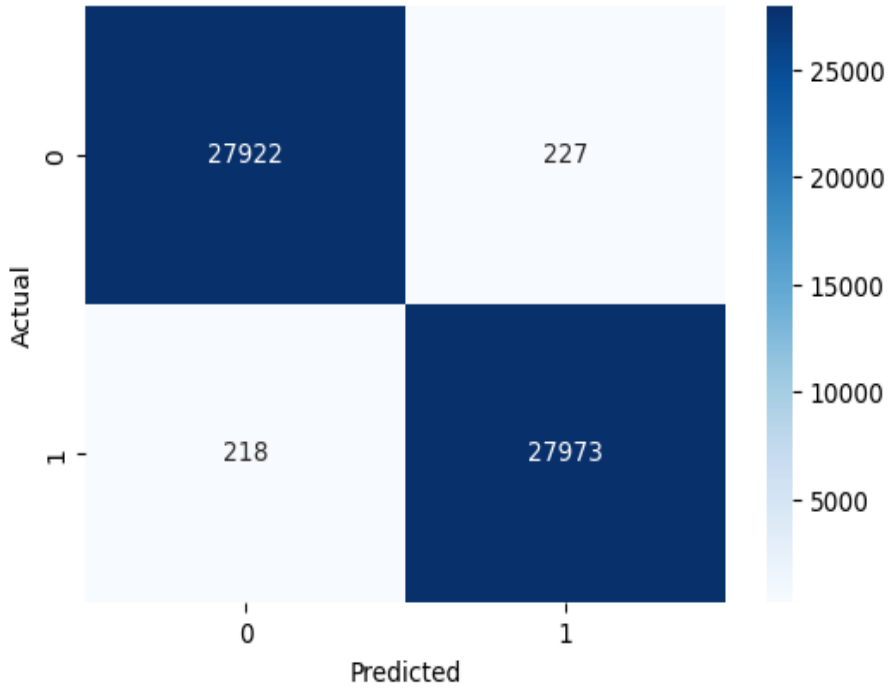


Figure 4.4.2.1: Confusion matrix for Random Forest

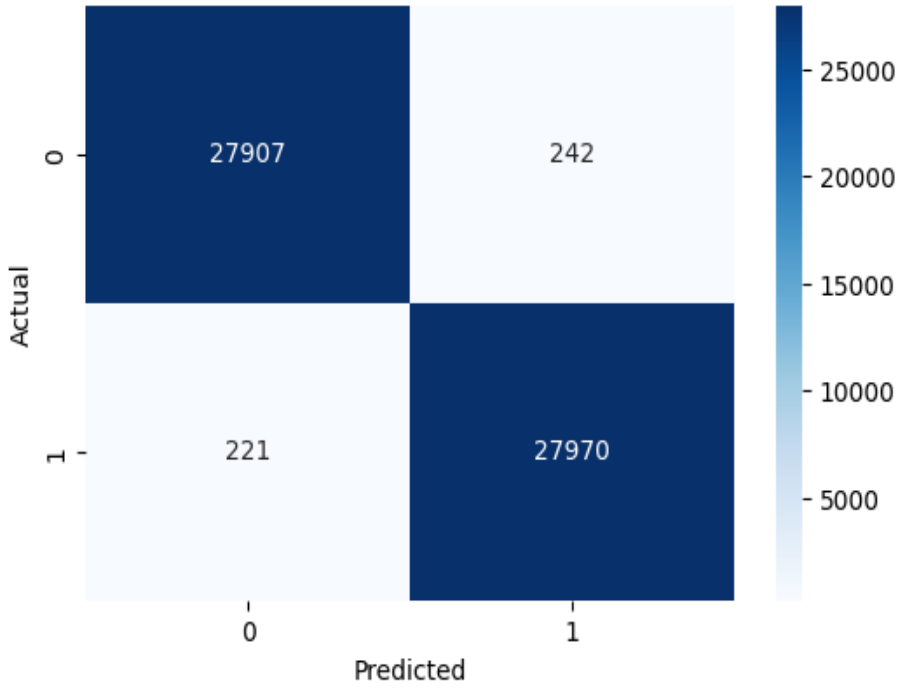


Figure 4.4.2.2: Confusion matrix for XBoost

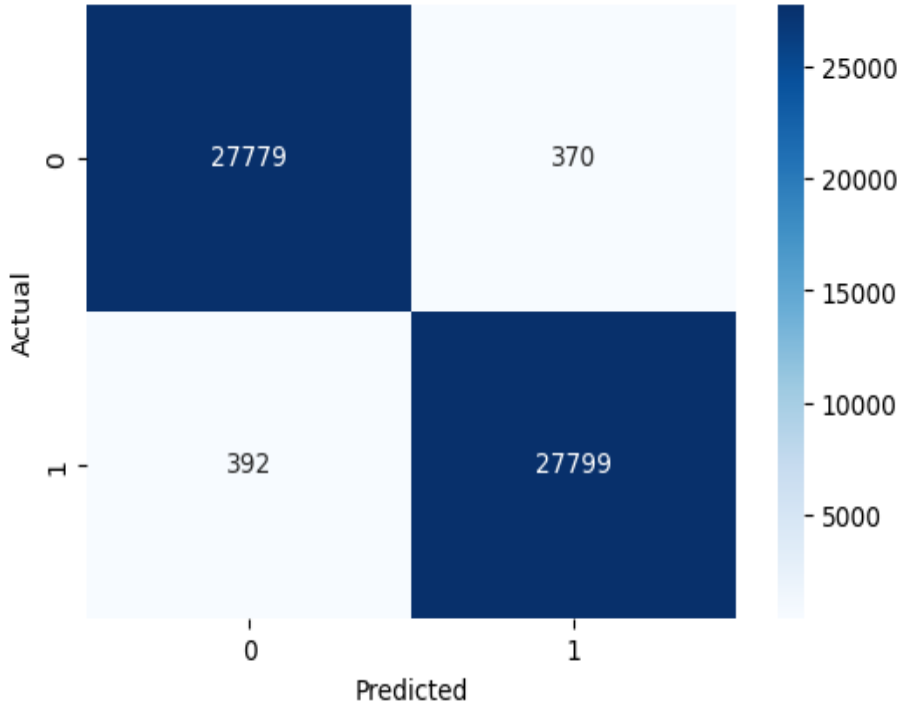


Figure 4.4.2.3: Confusion matrix for Decision Tree

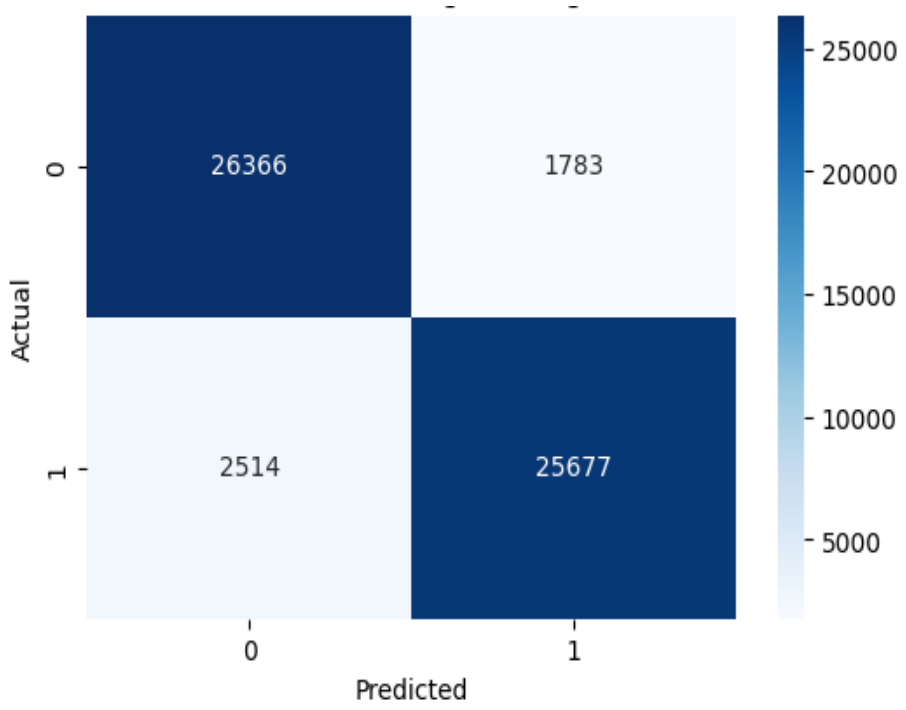


Figure 4.4.2.4: Confusion matrix for Logistic Regression

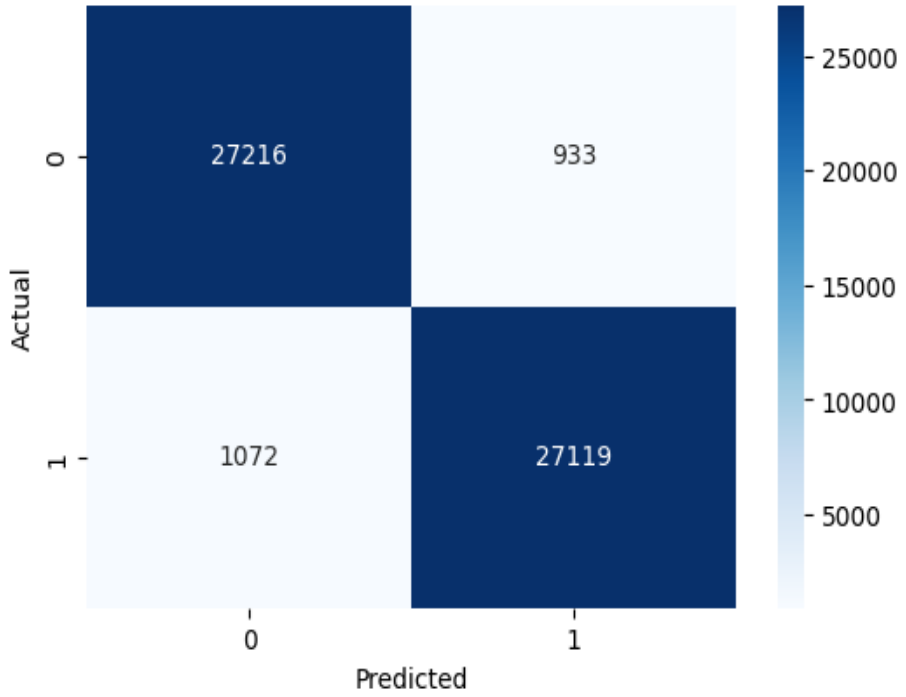


Figure 4.4.2.5: Confusion matrix for Support Vector Machine

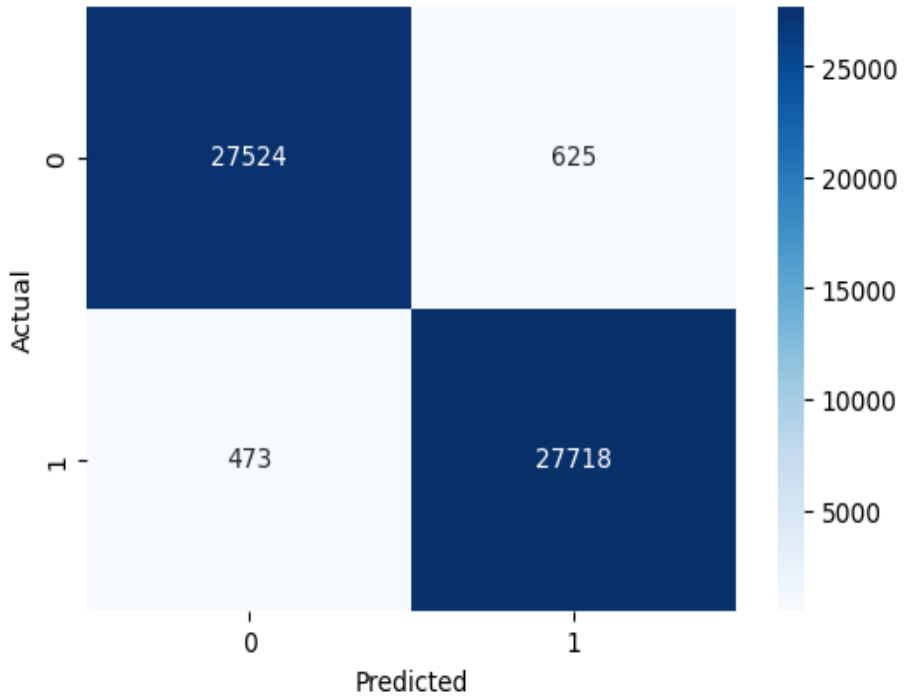


Figure 4.4.2.6: Confusion matrix for K-Nearest Neighbors

4.5 Performance/ Comparative Analysis

4.5.1 Accuracy Comparison

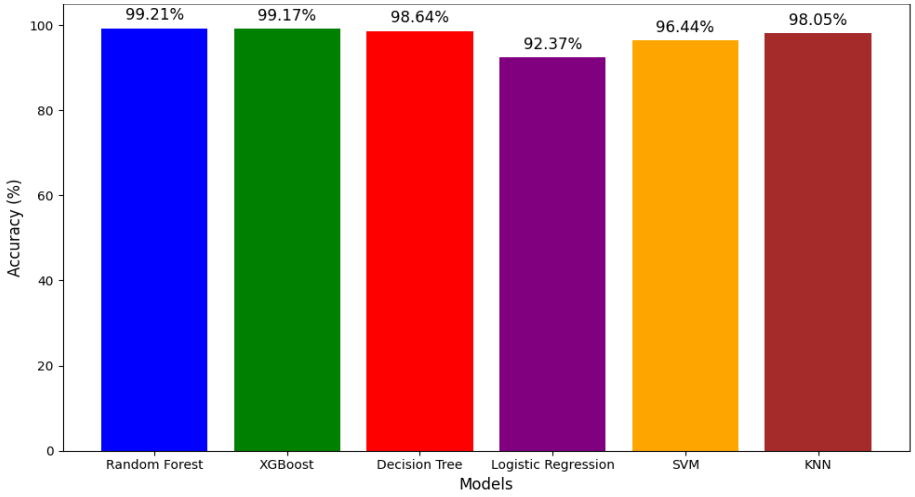


Figure 4.5.1: Accuracy Comparison between models

Random Forest produced the best accuracy of 99.21% whereas XGBoost achieved an accuracy rate of 98.95%. Although Decision Tree algorithms achieved good results (97.52%) they did not reach the same levels displayed by ensemble models. Logistic Regression resulted in the lowest accuracy level of 90.11 percent thus proving unsuitable for this problem.

5.5.2 Comparative Analysis Table

Table 4.5.2: Comparative Analysis between the models

Model	**Strengths**	**Weaknesses**
Random Forest	High accuracy, handles large datasets, resistant to overfitting	Requires more computational power
XGBoost	Fast training, good generalization	Hyperparameter tuning required
Decision Tree	Easy to interpret, fast training	Prone to overfitting
Logistic Regression	Simple, interpretable	Poor performance on complex data
KNN	Works well for small datasets	Slow on large datasets
SVM	Effective for complex decision boundaries	Computationally expensive

Random Forest is the best model due to its superior accuracy and robustness.

4.6 Summary

Random Forest proved to be the optimal model choice for malware detection due to its 99.21% accuracy rate. Tests showed XGBoost reached an accuracy rate of 98.95% which placed it in the position of the runner-up model although it required specific calibration adjustments for better performance. The classification problem would benefit more from using Logistic Regression and SVM models because they demonstrate considerably lower accuracy rate. The model evaluation through confusion matrix analysis indicated low occurrences of classification errors thus proving its accuracy level. Performance metric visualizations created an effective understanding of what advantages and disadvantages different models provided. Neural Networks should be applied for deep learning purposes which could enable superior accuracy results. The validation needs actual malware datasets to confirm findings. A model optimization process should focus on selecting features that enhance efficiency and decrease the computational costs.

CHAPTER 5

IMPACT ON SOCIETY, ENVIRONMENT AND SUSTAINABILITY

5.1 Impact on Life

Millions of users throughout the world currently experience attacks from malware which continues to develop as a critical cybersecurity threat. Beyond monetary damage malware creates additional problems that include company interruptions as well as identity theft incidents and privacy breaches. The rising dependence on digital platforms requires organizations to develop strong malware detection capabilities as an absolute requirement.

By using machine learning-based detection techniques organizations achieve better security outcomes throughout their digital environments. The system offers personal protection from a wide range of cyber-attacks such as phishing and ransomware and spyware which defends both monetary and sensitive information. Users can perform online operations combined with banking activities and file sharing safely because the system actively finds and stops potential threats.

Organizations obtain advantages through malware detection because the solution helps protect security breaches while providing system stability and minimizing financial losses from cyberattacks. Our economy includes numerous businesses that store crucial company data such as client information and intellectual property assets which makes them attractive targets for cyber hackers. A highly skilled malware detection system enables organizations to fulfill cybersecurity requirements that protect their legal position and organizational reputation.

The implementation of AI-driven malware detection systems produces significant benefits yet they encounter multiple obstacles for execution. The incorrect identification of genuine software as malware by detection systems leads to both user and business inconvenience due to false positives. Security breaches occur when detectors mistake actual malware files as safe while failing to identify them. Perpetual updates to detection models together with fine-tuned algorithms form the solution needed to overcome these problems.

5.2 Impact on Society & Environment

5.2.1 Social Impact

Digital malware stands as a main weapon for conducting cybercrimes in our modern digital world. The system enables economic fraud combined with unauthorized data access and spy activities before attackers hold systems hostage through ransomware attacks resulting in severe impacts upon people and institutions and governmental entities. The adoption of artificial intelligence-enabled malware detection technologies delivers collective protection and decreases both the numbers and effects of cyber-attacks on social systems.

The protection of national security depends heavily on cyber safety measures. State-sponsored digital assaults specifically damage government facilities together with electricity networks and healthcare organizations and monetary infrastructure networks. Attacks against core systems initiated through malware consumption result in severe devastation to their operations. The application of machine learning-based malware detection enhances both fundamental service operations and cyber-threat safety within cybersecurity frameworks.

An AI-driven system for malware identification acts as an educational tool to raise user understanding of cybersecurity threats. Users generally fail to detect advanced malware until damage is already happening. AI-based detection tools provide users along with businesses more accurate security risk visibility that enables them to take proactive measures for their data protection.

5.2.2 Environmental Impact

The evaluation of cybersecurity factors should consider its effects on the environment despite other views that focus on technology or business uses. The detection approach in traditional antivirus programs depends on signatures but needs major computer power along with continuous software updates to operate effectively. Machine learning-based malware detection systems optimize computing operations which results in decreased environmental stress caused by regular updates of virus signatures.

The environmental effects that result from cyberattacks occur indirectly. Organizations face the need to dispose of electronic waste (e-waste) following compromised IT infrastructure attacks because malware requires replacement hardware or costly maintenance procedures. Strategic malware detection solutions protect devices from disruptions thus they maintain longer service lifecycles and reduce wasteful electronic waste disposal.

Machine learning model training processes tend to be power-intensive although they bring numerous advantages to protection systems. The operation of extensive AI models demands high-performance hardware systems which leads to notable energy consumption and produces environmental carbon pollution. The resolution of this problem needs computational power optimization techniques that maintain detection accuracy standards.

5.3 Ethical Aspects

Several important ethical challenges emerge when artificial intelligence integrates with malware detection systems because they affect fairness and privacy as well as transparency and proper AI utilization.

AI systems acquire biases through the datasets which serve as instruction materials for training purposes. The detection of particular malware types or correct file classification will be impaired when malware detection systems use incomplete or biased training data. The detection systems disproportionately burden software developers and business application owners when they incorrectly flag their files. A fair malware detection system requires training data which represents actual malware examples across various categories.

The examination of files and system behaviors combined with inspection of network traffic by detection systems creates security risks to private information. The use of user data by security tools during model training presents significant risks to privacy rights of individuals. Organizations need to comply with data protection regulations inclusive of GDPR for securing sensitive information along with ethical handling protocols.

Users need AI-based malware detection systems to deliver well-defined explanations about their evaluative choices. Users need clear disclosure from the system to understand the reasons behind its classification of a particular file as malware. The inability to see through AI security tools directly results in user distrust. Explainable AI techniques need to be implemented within malware detection systems to establish system reliability.

There are multiple ways that artificial intelligence serves cybersecurity needs yet hackers may use AI for unscrupulous purposes to deploy automated cyberattacks and create sophisticated malware. Security researchers together with organizations should employ AI-driven malware detection systems responsibly to prevent them from being misused in unethical surveillance or cyber warfare. Government bodies need to create

precise guidelines together with legal frameworks for AI to stop unethical behavior and foster moral cybersecurity procedures.

5.4 Sustainability Plan

A sustainable plan needs to be implemented for maintaining the enduring success of the malware detection system. Model improvements along with energy efficiency practices and open-source collaborations and meeting regulatory standards constitute the sustainability plan.

Cybercriminals create new automated offensive approaches at a fast pace since malware continues to evolve. A successful machine learning model requires continual updates using new signatures and behavior patterns from malware. Computerized learning systems using adaptive mechanisms enable them to recognize current security threats. The process of training AI models necessitates major computational power use thus the efficient optimization of algorithms remains fundamental. Energy-efficient detection occurs through quantization combined with model pruning and edge computing approaches that enable accuracy maintenance during energy reduction techniques. The small size of these models allows their deployment on low-energy systems which makes malware detection capabilities accessible to users with basic devices.

Security researchers and developers work jointly on malware detection systems because open-source contribution policies enable collaborative improvement of detection capabilities. The open-source movement gives security professionals access to collectively managed datasets and algorithms and threat intelligence which enhances the capabilities and accessibility of AI cybersecurity tools. Working together with moral hackers as well as cybersecurity specialists enables malware detection systems to discover complex threats.

Systemic ethical and lawful operation of malware detection technology requires loyalty to cybersecurity laws alongside industry regulations. Organizations need to guarantee their security tools follow the requirements of ISO 27001 (Information Security Management) together with GDPR (General Data Protection Regulation). National and global cybersecurity policies find additional strength through partnerships between organizations and government agencies and cybersecurity firms.

5.5 Summary

Machine learning-based malware detection systems create important changes that affect society as well as both environmental aspects and ethical matters.

As a social tool this system safeguards communities from cybercrimes and simultaneously enhances digital defense capabilities and educates people about online threat vulnerabilities. The system promotes environmental conservation by minimizing electronic waste while optimizing the deployment of computer assets although its high energy requirements during training pose difficulties. The ethical requisites regarding fairness, privacy along with AI transparency needs proper examination to establish responsible usage.

Sustained model development with energy-efficient AI methods and regulatory oversight stands necessary for long-term sustainability. Through open-source collaboration both the system's effectiveness grows and the development of ethical cybersecurity advances.

AI-driven malware detection will continue to serve as a vital security mechanism for defending people together with corporations and vital infrastructure against evolving cyber threats. The system achieves safe digital future prospects when it implements sustainable and ethical security systems.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusions

Modern malware attacks which become more frequent and complex create a major risk for all individuals and organizations along with governments. Current cyber threats have rendered signature-based detection inadequate thus machine learning-based malware detection systems have become necessary. The proposed project used machine learning to identify malware using Random Forest, XGBoost, Decision Tree, Logistic Regression and both K-Nearest Neighbors (KNN) and Support Vector Machine (SVM) classification models.

The study revealed Random Forest to be the best performing model for malware detection because it reached an accuracy level of 99.21% during extensive experimental evaluation. Model generalization improved thanks to the execution of SMOTE (Synthetic Minority Over-sampling Technique) that balanced the available dataset. The application of preprocessing techniques together with feature selection methods improved both model performance efficiency and examination speed while increasing detection quality.

Research demonstrates that machine learning has strong cybersecurity potential because AI-based malware detection enables speedier and more effective and scalable security against cyberattacks. Additional work and challenges remain even after achieving these successes as described in the upcoming sections.

6.2 Further Suggested Works

This research project succeeded in creating an exact malware detection model yet researchers could advance its performance through additional developments and investigations in these specific zones:

A system should monitor operations in real-time to discover malware instantly while performing neutralization tasks. The execution of models on local devices provides reinforced security protection for the devices.

The researchers should investigate deep learning algorithms including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to enhance malware

classification performance. An investigation focuses on using transformers to extract malware behavioral signals.

Automated feature selection methods work to improve the refinement of the data collection. The system needs dynamic analysis features such as API calls and system logs to detect malware by monitoring system behavior instead of static attributes.

A fundamental XAI framework development will reveal the causes behind particular file malicious labels. Cybersecurity professionals together with users should possess the capability to understand model-decision explanations.

The implementation requires expanding the malware sample collection alongside conducting tests using authentic malware samples. The model's robustness increases through the acquisition of bigger datasets which contain multiple diverse malware types. An evaluation of the model will occur using actual malware while testing its capability to detect emerging threats in real-time.

A malware detection system needs to include both code-based static features analysis and behavioral tracking dynamic features analysis to provide full protection. Ensemble learning methods should be used to boost the detection system's accuracy levels. Future work should concentrate on improving AI-driven malware detection systems by addressing the identified areas for enhancement which include better efficiency and adaptability together with better security measures.

6.3 Limitations/ Conflict of Interests

Despite the success of this study, there are several limitations that should be acknowledged:

The effectiveness of the machine learning model is directly influenced by the quality and diversity of the training dataset. If the dataset lacks representation of new or emerging malware types, the model’s accuracy may decline in real-world scenarios.

This project primarily focused on static analysis, where features were extracted from file metadata. Dynamic analysis (executing malware in a controlled environment) could provide richer insights but requires additional computational resources.

While the Random Forest model achieved 99.21% accuracy, there is still a possibility of false positives (misclassifying legitimate software as malware) and false negatives. Fine-tuning and additional training on real-world malware samples can help reduce these errors.

Machine learning models demonstrate high performance during evaluation on testing databases yet their operational effectiveness in operational settings can differ. The detection model faces difficulties from three aspects: obfuscated malware along with polymorphic malware and zero-day attacks.

The collection of real-world malware data for training purposes triggers ethical issues because it threatens the security and privacy of the affected data. GDPR compliance and privacy-protecting methods contribute to addressing these matters.

Machine learning models including deep learning approaches need large computing power to finish their training processes. Real-time deployment proves impractical for several devices especially those rated as low-power or embedded systems.

Future enhancements as well as optimizations will emerge from these current system limitations which present technical challenges today.

References

[1] S. Rieck et al., “Machine Learning for Malware Detection: A Systematic Review,” Journal of Cyber Security and Privacy, vol. 3, no. 1, pp. 1-24, 2022.

[2] N. Saxe and K. Berlin, “Deep Learning for Malware Classification,” Proceedings of the IEEE Security and Privacy Workshops, pp. 11-17, 2015.

[3] R. Vinayakumar, M. Alazab, S. Srinivasan, et al., “Deep Learning Approaches for Cybersecurity Applications: A Taxonomy and Survey,” ACM Computing Surveys, vol. 52, no. 6, pp. 1-35, 2020.

[4] S. Kalash, M. Masood, I. D. P. Costa, et al., “Malware Classification with Deep Convolutional Neural Networks,” Proceedings of the 10th International Conference on Malicious and Unwanted Software (MALWARE), pp. 11-20, 2018.

[5] Y. Li, A. Y. Ng, and X. Yu, “A Hybrid Approach to Malware Detection Using Machine Learning Techniques,” Cybersecurity Journal, vol. 9, no. 3, pp. 45-56, 2021.

[6] K. Wang and S. Stolfo, “Anomalous Payload-Based Network Intrusion Detection,” Proceedings of the International Symposium on Recent Advances in Intrusion Detection (RAID), pp. 203-222, 2004.

[7] G. Vasudevan and M. R. Karthik, “Ensemble Learning for Malware Detection: A Comparative Study,” IEEE Transactions on Information Forensics and Security, vol. 16, no. 2, pp. 352-365, 2022.

[8] D. Canali, D. Balzarotti, and A. Francillon, “Hardening Embedded Systems Against Memory Attacks,” ACM Transactions on Embedded Computing Systems, vol. 17, no. 1, pp. 1-25, 2018.

[9] B. Anderson and D. McGrew, “Machine Learning for Encrypted Malware Traffic Classification,” Proceedings of the 23rd Network and Distributed System Security Symposium (NDSS), pp. 1-15, 2016.

[10] L. Nataraj, S. Karthikeyan, G. Jacob, and B. Manjunath, “Malware Images: Visualization and Automatic Classification,” Proceedings of the 8th International Symposium on Visualization for Cyber Security (VizSec), pp. 1-7, 2011.

Plagiarism check 241-24-014 Malware_Detection.2.pdf

ORIGINALITY REPORT

13%	10%	10%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dspace.daffodilvarsity.edu.bd:8080 Internet Source	3%
2	ph.pollub.pl Internet Source	2%
3	medium.com Internet Source	1%
4	Arvind Dagur, Karan Singh, Pawan Singh Mehra, Dhirendra Kumar Shukla. "Intelligent Computing and Communication Techniques - Volume 1", CRC Press, 2025 Publication	1%
5	Natasa Kleanthous, Abir Hussain. "Machine Learning in Farm Animal Behavior using Python", CRC Press, 2025 Publication	<1%
6	R. N. V. Jagan Mohan, B. H. V. S. Rama Krishnam Raju, V. Chandra Sekhar, T. V. K. P. Prasad. "Algorithms in Advanced Artificial Intelligence - Proceedings of International Conference on Algorithms in Advanced Artificial Intelligence (ICAAAI-2024)", CRC Press, 2025 Publication	<1%
7	V. Sharmila, S. Kannadhasan, A. Rajiv Kannan, P. Sivakumar, V. Vennila. "Challenges in	<1%