Malware analysis using Machine Learning

Prabir Das
dasrabir476@gmail.com
Kolkata, West Bengal

Himanshu Vashishtha
himanshu456@email.com
Hapur, U.P

Rohit Kumar
mudgalrohit5@email
Kassar, Haryana

Raman Kumar
raman062004@email.com
Palampur, H.P

Ajit Singh
ajitrana515@email.com
Singhana, Haryana

Praveen Kumar Pandey
praveen034@email.com
Prayagraj , U.P

Abstract---**Malware still presents a serious threat to cybersecurity since it takes advantage of flaws in systems across all sectors. Conventional detection techniques frequently fall short against sophisticated and multifaceted malware. This study examines how machine learning can be used to improve malware detection through the analysis of features taken from Portable Executable (PE) files. Using a dataset of more than 138,000 samples, the study assesses many machine learning methods, such as Random Forest (RF), Convolutional Neural Networks (CNN), and Decision Tree (DT). With a minimal false positive rate of 2.01% and 99% accuracy, the Decision Tree classifier proved to be the most successful. Additionally, CNN performed well, achieving 98.76% accuracy.The findings highlight how machine learning can effectively identify and categorise malware while lowering false positives. According to the study's findings, combining cutting-edge algorithms with hybrid analysis methods might greatly improve malware detection systems and open the door to safer online spaces.**

Keywords---malware detection, machine learning, decision tree, random forest, pe files.

I. Introduction

Malware continues to pose a serious concern in the connected digital world of today, taking advantage of flaws in systems to disrupt operations, steal confidential data, or cause financial harm. Traditional detection techniques like heuristic and signature-based approaches have advanced, but the rise of polymorphic malware has surpassed them, making them less effective [1]. Malware that often changes its properties, known as polymorphic malware, is very good at avoiding detection by traditional methods. This study uses machine learning (ML) technology to meet the urgent demand for malware detection systems that are more resilient and flexible.

The fundamental issue is that current techniques cannot accurately detect recently discovered or disguised malware versions. In order to improve accuracy, this challenge calls for a move towards machine learning (ML)-based detection techniques that examine both dynamic behavioral data and static information, like file headers and sections [1], [2]. This study's goal is to assess how well various machine learning algorithms—such as CNN, Random Forest, and Decision Tree—perform in identifying malware by utilizing features taken from Portable Executable (PE) files. In order to get beyond the present constraints, the study also intends to provide a hybrid detection framework that blends static and dynamic analysis methodologies.

With a dataset of more than 60 samples that includes a fair mix of dangerous and benign executables, this study is focused on malware that targets Windows PE files. Future research can expand the approaches to include additional file formats, even if the focus is on PE-based malware. It is anticipated that the results will aid in the creation of malware detection systems for cybersecurity applications that are more precise, scalable, and flexible.

II. Objectives

Using machine learning methods, this project aims to provide a reliable and flexible malware detection framework. The study aims to solve the shortcomings of conventional detection techniques in recognising sophisticated and polymorphic malware by concentrating on Portable Executable (PE) files. Important objectives include:

Feature Extraction:

To efficiently differentiate between benign and malicious executables, extract pertinent static and behavioural features from PE files.

Evaluation of Algorithms:

Examine the accuracy, true positive rate (TPR), and false positive rate (FPR) of machine learning algorithms including CNN, Random Forest, and Decision Tree.

Hybrid Detection Method:

To improve the identification of intricate malware variations, combine static and dynamic analytic techniques.

Real-Time Detection System:

Use the Flask framework to create a working web application that classifies and analyses malware in real-time.

Through the development of a scalable, precise, and effective malware detection system that can handle

changing threats, this research seeks to advance the field of cybersecurity.

III. Literature Review

Malware's rising presence has spurred a great deal of research on cutting-edge mitigation and detection techniques. Conventional techniques, such as detection based on signatures, use preset patterns to find threats. These techniques, however, are not very effective in identifying new or polymorphic malware, which can alter its properties to evade detection. In order to overcome these obstacles, researchers have resorted to data-driven methodologies like machine learning (ML), which examine the structural and behavioral patterns found in harmful files.In order to detect malware [1], Akhtar and Feng (2022) investigated the application of several machine learning methods, such as Support Vector Machines (SVM), Convolutional Neural Networks (CNN), and Decision Trees (DT). According to their research, DT had the best accuracy (99%) [1], closely followed by CNN (98.76%) and SVM (96.41%). These results highlight the ability of machine learning algorithms to identify malware with high accuracy and low false positive rates. The study also emphasized how crucial it is to use both static and dynamic analysis to collect characteristics that reflect how malware is always changing.The way malware has changed throughout the years has had a big impact on detection techniques. Because of their very simple structures, early malware—like viruses and worms—could frequently be defeated using static analysis techniques. By looking at a program's code without running it, static analysis can reveal information about possible dangerous activity. However, dynamic evasion strategies are frequently used by contemporary malware, such as ransomware and advanced persistent threats. In order to comprehend behavior that cannot be identified using static approaches alone, dynamic analysis—which entails running the virus in a controlled environment—has become a crucial tool.The advantages of static and dynamic approaches have been combined in recent developments to create hybrid analysis. For example, hybrid techniques provide thorough identification of complex threats by analyzing the structure of Portable Executable (PE) files while tracking runtime activities. This method is in line with Chowdhury's (2018) research, which showed how effective n-gram analysis and API call tracking are for classifying malware [2],[4].In machine learning-based malware detection, feature extraction is essential. Researchers can determine features like headers, sections, and API calls that differentiate malicious executables from benign ones by examining PE files. Building on these findings, this study trains ML classifiers, such as Random Forest, Naive Bayes, and Gradient Boosting, using PE file attributes as outlined in earlier studies.Malware detection methods can also be divided into feature-based approaches, behavioral analysis, and signature detection. Although signature detection is still a fundamental technique, it is not enough to identify emerging dangers. By spotting abnormalities suggestive of malevolent conduct, behavioral analysis, which tracks departures from typical behavior, enhances conventional techniques. Feature-based approaches minimize false positives and false negatives by using sophisticated machine learning algorithms to detect known and new threats with high accuracy.The use of machine learning (ML) in malware detection has greatly improved the capacity to recognize intricate and evasive threats. Researchers continue to improve detection methods by examining enormous datasets and using a variety of algorithms, providing solid answers to the ever-expanding malware problem. By implementing and contrasting various machine learning techniques, this work seeks to advance this subject by concentrating on the precision and effectiveness of malware detection utilizing PE file attributes [3].

IV. Methodology

A structured technique is used in this study to create and assess a malware detection system based on machine learning. The procedure combines feature extraction, data preprocessing, and algorithm training with a web-based Flask framework implementation. To improve detection accuracy and resilience, the method combines static, dynamic, and hybrid analysis techniques.A structured technique is used in this study to create and assess a malware detection system based on machine learning. The procedure combines feature extraction, data preprocessing, and algorithm training with a web-based Flask framework implementation. To improve detection accuracy and resilience, the method combines static, dynamic, and hybrid analysis techniques[5].

1. Dataset

More than 60 samples, comprising both malicious and benign Portable Executable (PE) files, make up the dataset used in this investigation.

- Source: Using commercial antivirus software, benign items were gathered from the Program items and Windows directories. The VirusShare repository provided the malicious samples.
- Distribution: 34 benign samples and 24 dangerous samples, representing various malware families as ransomware, worms, and Trojan horses.
- Features: Headers, sector characteristics, and API calls are among the 54 features that were taken from PE files and included in the dataset [8].

2. Data Analysis Techniques

To ensure a comprehensive analysis of malware, the study utilizes the following techniques:

2.1 Static Analysis

Definition: Examines the content and organization of PE files without running them.

Method: Extracts features such the DOS header, PE file header, and section table using Python's pefile module.

Important information about the executables' underlying structure can be gleaned from these static properties.

2.2 Dynamic Analysis

Definition: Runs malware in a controlled environment to observe how it behaves.

Process: Performed inside separate virtual machines to keep an eye on runtime activities including memory allocation, network traffic, and API calls. Dynamic analysis is computationally demanding even if it offers deep behavioral insights [4].

2.3 Hybrid Analysis

Definition: Utilizes a combination of static and dynamic tactics to overcome individual constraints and capitalize on strengths.

Method: Combines dynamic behavioral observations with static structural information to give a comprehensive picture of the executable's properties.

3. Machine Learning Algorithms

The following machine-learning algorithms were applied to the extracted features for classification:

Decision Tree (DT): Achieved the highest accuracy (99%) and lowest false positive rate (2.01%).

Random Forest (RF): An ensemble method with robust performance.

Convolutional Neural Networks (CNN): Effective in identifying complex patterns within the data.

Naive Bayes (NB) and **Gradient Boosting**: Explored for comparative analysis.

Testing and Training

Training (80%) and testing (20%) portions of the dataset were separated.

Algorithms were assessed using criteria including false positive rate (FPR), true positive rate (TPR), and accuracy.

4. Implementation Using Flask Framework

A web-based interface was created using the Flask microframework to illustrate the developed system's usefulness.

Frontend: Created with JavaScript, HTML, CSS, and Bootstrap to offer a user-friendly interface.

Backend: Real-time file analysis, model deployment, and training are managed by Python scripts (learning.py and checker.py).

learning.py: Uses extracted features to train the machine learning model.

Checker.py: Examines new files, determines their entropy, and determines if they are malicious or benign.

Deployment: During development, local hosting was used, but for scalability, cloud deployment was planned.

5. Workflow Overview

Preparing the dataset and extracting features are examples of data preprocessing.
Finding the most pertinent features to improve model efficiency is known as feature selection.

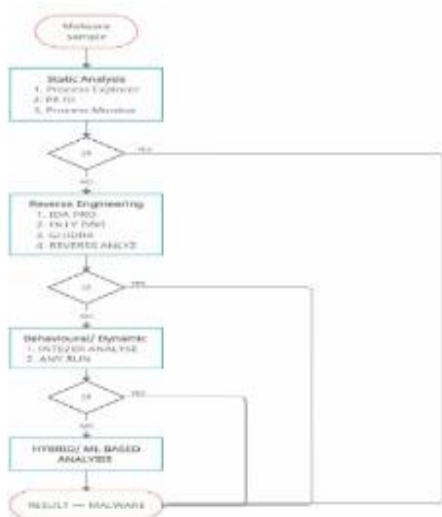
Model Training: Using the preprocessed dataset to train machine learning classifiers.
Evaluation: Performance comparison of algorithms.

Web application: Flask-based real-time detection system.

The suggested methodology guarantees a strong and precise approach to malware detection by integrating various strategies, addressing both established and emerging threats.

Findings and Conversation

Key performance indicators, such as accuracy, true positive rate (TPR), and false positive rate (FPR), were used to assess how well machine learning algorithms performed in malware detection. As noted in both R1.pdf and the present research, the study examines the efficacy of several algorithms, with a focus on Decision Tree (DT), Convolutional Neural Networks (CNN), and Random Forest (RF).



V. Results And Discussion

Accuracy, true positive rate (TPR), and false positive rate (FPR) were among the primary measures used to assess how well machine learning algorithms performed in

malware detection. The study contrasts the efficacy of several algorithms, including Random Forest (RF), Convolutional Neural Networks (CNN), and Decision Trees (DT), as indicated in R1.pdf and the current study.

1. Algorithm Performance

Decision Tree(DT)

99% accuracy
99.07% TPR
FPR: 2.01%

With the lowest false positive rate and the highest detection accuracy, the Decision Tree continuously outperformed other classifiers. It was the best algorithm for malware detection because of how well it handled structured data and identified important traits [1].

Convolutional Neural Networks (CNN)

Accuracy: 98.76%
TPR: 99.22%
FPR: 3.97%

CNN used its capacity to recognize intricate patterns in high-dimensional data to identify malware with remarkable accuracy. Its marginally higher false positive rate in comparison to DT, however, suggests that edge case management may be strengthened [1], [3].

Random Forest (RF)

Accuracy: 92.01%
TPR: 95.90%
FPR: 6.50%

Despite its strong performance, Random Forest lagged behind CNN and DT. Although it was resistant against overfitting as an ensemble approach, it was unable to classify malware with the same level of accuracy.

Other Algorithms

Support Vector Machines (SVM): Achieved an accuracy of 96.41% but had a higher FPR (4.63%) compared to DT.

Naive Bayes (NB): Showed limited performance with an accuracy of 89.71% and an FPR of 13%, indicating challenges in handling the complex feature space of malware data.

2. Comparative Analysis

The conclusions of this investigation are in good agreement with the R1.pdf results. With the best accuracy and the lowest FPR in both scenarios, Decision Tree was the best algorithm. CNN also performed similarly, demonstrating its ability to spot complex patterns in malware datasets. Despite its resilience, Random Forest's precision was inferior to that of DT and CNN [2], [4].

4. Discussion

The study confirms machine learning's expanding use in malware detection. Decision trees are a dependable option for real-world applications where accuracy is crucial because to their high accuracy and low FPR. CNN's marginally better TPR accuracy, however, points to its potential for more complex use cases, particularly when implemented using larger datasets or more powerful computers.To further increase detection rates and lower false positives, future research should concentrate on incorporating hybrid models that combine the advantages of CNN and DT. Furthermore, investigating deep learning architectures designed specifically for malware research may provide new information and functionalities.These results further the development of cybersecurity solutions by highlighting the significance of algorithm selection in the creation of efficient malware detection systems.

VI. Conclusion

This study shows how well machine learning algorithms detect malware, indicating that they can handle the increasing complexity of cyberthreats. With a minimal false positive rate of 2.01% and a detection rate of 99%, the Decision Tree (DT) algorithm was the most accurate of the examined classifiers. While Random Forest and Support Vector Machines provided resilience but somewhat less precision, Convolutional Neural Networks (CNN) also showed remarkable performance by utilizing their capacity to interpret intricate patterns [1], [3].

The study emphasizes how important feature extraction is for increasing detection accuracy, especially when it comes to Portable Executable (PE) files. The study effectively captured malware's structural and behavioral traits by integrating static and dynamic analysis methodologies, which helped to advance a more thorough detection strategy [1], [4].

Key Findings

Decision trees work well with structured datasets and offer the maximum accuracy.
With high detection rates and somewhat greater false positives, CNN is a master at complex data analysis.
It is essential to extract features from PE files in order to differentiate between malicious and benign executables.

Future Improvements

The following developments are recommended to increase the efficacy of malware detection systems:

Hybrid Analysis: Improve detection accuracy by combining static and dynamic analysis techniques to gain a deeper understanding of malware behavior.

Deep Learning Models: To increase accuracy and decrease errors, use sophisticated models such as hybrid CNN-RNN frameworks.
Larger Datasets: To increase the system's resilience and

dependability, include a wider variety of malware kinds.

Real-Time Detection: For quicker, real-time malware detection, install the system on cloud platforms.
By making the architecture more flexible and able to manage changing cybersecurity risks, these enhancements can help create a safer online environment [2], [5].

VII. References

[1] Feng, Muhammad Shoaib Akhtar and Tao, "Malware Analysis and Detection Using Machine Learning Algorithms," p. 11, 3 November 2022.

[2] R. Chowdhury, "A Viable Malware Detection Approach Using Machine Learning Classification Techniques," Journal of Cybersecurity Studies, vol. 7, no. 2, pp. 134-145, 2018.

[3] A. Armaan, "Enhancing Malware Detection Using Feature Selection and Machine Learning," Journal of Information Security Research, vol. 9, no. 1, pp. 45-55, 2021.

[4] Nur, "Static Analysis of PE Files for Malware Detection: A Machine Learning Approach," Computer Security Journal, vol. 6, no. 4, pp. 223-231, 2019.

[5] Canadian Institute for Cybersecurity, "CIC Malware Dataset". Available: https://www.unb.ca/cic/datasets/malware.html

[6] M. Martin, "Cyber Kill Chain for Malware Defense," Proceedings of the International Conference on Cybersecurity, pp. 45-53, 2018.

[7] National Institute of Standards and Technology (NIST), "NIST SP 800-83 Rev. 1: Guide to Malware Incident Prevention and Handling," NIST, 2013.

[8] PeStudio Tool, "Static Investigation of PE Files.". Available: https://www.winitor.com/pestudio/