Journal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 15, Issue. 1, No.10 : 2024
ISSN : **1906-9685**

MALWARE DETECTION USING MACHINE LEARNING TECHNIQUES

[#1]**Mrs G. SUJATHA,** Associate Professor in Computer Science Department (CSE),
VIGNAN'S INSTITUTE OF INFORMATION TECHNOLOGY
VISAKHAPATNAM, INDIA
[#2]**HARSHA PRIYA UJJURU,** Student Department of Computer Science & Engineering,
[#3]**EMANDI CHANDRA MOULI,** Student Department of Computer Science & Engineering,
[#4]**KAKUMANI AISWARYA,** Student Department of Computer Science & Engineering,
[#5]**GADI GANESH,** Student Department of Computer Science & Engineering,
[#6]**LADAY SAMPATH VIGNESH,** Student Department of Computer Science & Engineering,

ABSTRACT: Our research introduces an efficient malware detection model that uses machine learning to tell apart safe and harmful executable files. We aim to combat the increasing malware threat with a straightforward yet effective approach. Our method combines two types of analysis: static and dynamic. We want to compare the results from both and create a better, hybrid method for more accurate detection. Our dataset contains malware and safe executable samples, which we use to train and test our model. We pay close attention to extracting useful information from file and section headers of portable executable files. For classification, we try different machine learning classifiers like Random Forest, KNearest Neighbors (KNN),Decision Tree . The standout performer is the Random Forest classifier, achieving an impressive 98% accuracy rate. This research shows that machine learning is effective in spotting malware and emphasizes the importance of picking the right classifier for the job. Our results support using Random Forest for real-world malware detection. Additionally, we compare static and dynamic analysis approaches, giving us insights into what each is good at and where they fall short. This can help in developing hybrid methods for even better malware detection. In summary, our study contributes to cybersecurity by offering a practical and efficient malware detection model that combines feature-based analysis with machine learning. The Random Forest classifier shows great potential for accurately distinguishing between safe and harmful executable files.

1. INTRODUCTION

Malware is a software that is specifically designed to disrupt, damage, or gain unauthorized access to a computer system.The continuous evolution and sophistication of malware pose a significant cybersecurity challenge in today's digital landscape. Traditional signature-based malware detection methods struggle to keep up with the ever-growing diversity of malware variants. As executable files continue to be a primary vector for malware infiltration, the need for robust and advanced detection mechanisms becomes paramount. This project focuses on the pivotal task of enhancing malware detection in executable files through the application of machine learning algorithms and leveraging the power of machine learning algorithms. we aim to develop a sophisticated system capable of effectively identifying malicious code patterns within executable files. By harnessing the analytical capabilities of machine learning, this project seeks to contribute to the ongoing efforts to fortify digital ecosystems against the ever-evolving landscape of malware threats.

2. REVIEW OF LITERATURE

[1] Muhammad Shoaib Akhtar et. al. [2022]: "Malware Analysis and Detection Using Machine Learning Algorithms". In this paper, they discuss the growing threat of polymorphic malware and the difficulty of creating a reliable detection system. The machine learning techniques they use include Random Forest, Naive Bayes, Decision Trees, Convolutional Neural Networks, Support Vector Machines, and a recommended method. The Canadian Institute for Cybersecurity's dataset is used to compare the accuracy, True Positive Rate, and False Positive Rate performance of different algorithms. The results demonstrate that convolutional neural networks, decision trees, and support vector machines effectively achieve high detection accuracy.

[2] Nighat Usman et al. [2021]: "Intelligent Dynamic Malware Detection using Machine Learning in IP Reputation for Forensics Data Analytics".The necessity for innovative cybersecurity methods to detect malicious IP addresses is discussed by the authors in their paper, They suggest a hybrid framework that blends data forensics, machine learning, cyber threat intelligence, and dynamic malware analysis. behavioral analysis is done using the Decision Tree approach. The usefulness of the suggested framework in lowering false alarms is demonstrated by the study's comparison of it with other machine-learning approaches and reputation systems that are currently in place.

[3] Ilker Kara et. al. [2022]: "Fileless malware threats: Recent advances, analysis approach through memory forensics and research challenges". This paper delivers insightful perspectives on the ever-evolving realm of fileless malware attacks. The author extensively dissects the intricacies of these sophisticated threats, accentuating the limitations of conventional detection methods. Kara introduces a state-of-the-art approach grounded in memory forensics, contrasting its efficacy with static analysis. The narrative underscores the inherent advantages of memory-based analysis, marking a qualitative leap in the detection of elusive file-less malware. Employing the real-world case study of the "Kovter" attacker, the paper exposes vulnerabilities in prevailing detection systems. While specific numerical metrics are absent, the qualitative merits of the proposed memory-based approach emerge as a promising stride in fortifying defenses against the challenges posed by fileless malware.

[4] Tal Tsafrir et al. [2023]: "Efficient Feature Extraction Methodologies for Unknown MP4-Malware Detection using Machine Learning Algorithms"In this paper, The authors delve into the security threats associated with MP4 files, shedding light on the vulnerabilities exploited by cybercriminals. The paper introduces innovative feature extraction methodologies tailored for the detection of unknown MP4 malware, presenting a comprehensive comparison of three distinct approaches. The configuration that outperforms the others demonstrates remarkable accuracy metrics, boasting an Area Under the Curve (AUC) of 0.9651, a True Positive Rate (TPR) of 0.976, and a remarkable False Positive Rate (FPR) of 0.0. The article effectively underscores the efficacy of the proposed methodologies in effectively discriminating between malicious and benign MP4 files, thus enhancing cybersecurity defenses.

[5] HemantRathore et al. [2023]: "Robust Malware Detection Models: Learning From Adversarial Attacks and Defenses" In this paper, The primary objective is to engineer malware detection models resilient to adversarial intrusions. Delving deep into the realm of cybersecurity, the authors meticulously craft a comprehensive framework. This framework amalgamates the prowess of various machine learning algorithms such as deep neural networks, decision trees, support vector machines, gradient boosting, and multiple iterations of random forests. By rigorously evaluating these frameworks and introducing potent countermeasures against adversarial threats, Rathore and his

team accentuate the paramount importance of adopting proactive strategies in the domain of malware detection and defense.

[6] AkshitKamboj et al. [2022]: "Detection of Malware in Downloaded Files Using Various Machine Learning Models".In this paper, They address the escalating menace of malware concealed in downloaded files. Their solution involves employing machine learning models such as XGBoost, AdaBoost, Gradient Boosting, Random Forest, Decision Tree, and Gradient Boosting. Notably, the Random Forest Classifier outshines others with an exceptional accuracy rate of 96.99%. This outstanding performance underscores its efficacy in providing a resilient defense against malware threats lurking within downloaded files. The study showcases the significance of leveraging advanced machine learning techniques for achieving superior accuracy in malware detection, ensuring enhanced cybersecurity measures.

3. PROPOSED SYSTEM

Malware must be analyzed in order to understand its content and actions. Malware analysis is the process of identifying the functionality of malware and the answers to the questions listed below. How malware works, which PCs and applications are impacted, what data is corrupted or stolen, and so on. Static, dynamic, and hybrid analysis are the three basic malware analysis approaches.

Static Analysis: Static analysis is software analysis that is conducted without actually running the programme . To do static analysis, many techniques are used. Some are based on the binary file's properties, such as extracting byte code sequences from the binary, extracting opcode sequences after disassembling the binary file, extracting control flow graphs from assembly files, extracting API calls from the binary, and so on. Each represents a feature set, and any one or a combination of them is utilized to identify malware.

Dynamic Analysis: Dynamic analysis is software analysis conducted while the application is running. API calls, system calls, instruction traces, taint analysis, registry changes, memory writes, and other information can be retrieved by dynamic analysis.This sort of study is often carried out in a sandbox environment to prevent malware from infecting production systems. Dynamic analysis requires more resources and has a higher cost.

Hybrid analysis: The hybrid analysis approach combines static and dynamic analysis techniques. Because of the multi-path execution, dynamic analysis might be time-consuming. Static analysis can be used to identify the path of execution for dynamic analysis, increasing accuracy and efficiency.This analysis approach was developed to address limitations in both static and dynamic analysis techniques. It begins by examining any malware code's signature and then combines it with other behavioral pattern factors to improve malware analysis. As a result, it solves both the static and dynamic analysis techniques' shortcomings. This improves the capacity to appropriately detect dangerous software. At the same time, this analytical approach possesses nearly all of the strengths of both static and hybrid techniques.

"The proposed system aims to revolutionize malware detection by leveraging the capabilities of machine learning algorithms. Traditional methods, while effective to some extent, often struggle to keep pace with the dynamic and sophisticated nature of modern malware. Our solution incorporates advanced machine learning techniques to enhance the accuracy, adaptability, and efficiency of malware detection. By employing algorithms such as Random Forest, Support Vector Machine, Ada Boost, K Nearest Neighbors, and Naive Bayes, we seek to create a comprehensive and proactive defense mechanism against a wide array of malware threats. The proposed system will not only identify known malware signatures but will also excel in detecting new and previously unseen variants, addressing the limitations of traditional signature-based approaches. Additionally, the system will conduct a thorough behavioral analysis, monitoring the actions of software to identify anomalies indicative of potential malware. This holistic approach is designed to minimize false positives and negatives, offering a more reliable and robust defense against the ever-evolving landscape of cyber threats. Through the proposed system, we aim to provide a cutting-edge solution that significantly elevates the cybersecurity posture of computer systems, ensuring a safer and more secure digital future."

Advantages of proposed system:

- Machine learning Adaptability.
- Enhanced Accuracy.
- Polymorphic Malware Resilience.
- Continuous Learning.

4. METHODOLOGY

(i)Dataset collection: In this work, a data set is used to classify malware with PE headers. These datasets are built with the header field values of the PE file. These data sets include some features regarding portable executable file format like image headers and file headers and optional headers. Include some information about the portable executable file in the static dataset. Image headers, file headers, and section headers are among the characteristics derived from portable executable file headers. The file headers include information about the operating system that will be used to run the executable. Optional headers guarantee that the entry point is executable. The data is contained in the section headers. Hence uses 31 file header features, 29 operational header features, and 19 section header features in this dataset. Include some information about the portable executable file in the static dataset. Image headers, file headers, and section headers are among the characteristics derived from portable executable file headers. The file headers contain information about the operating system used to run the executable. Optional headers guarantee that the entry point is executable.The hybrid model has collected some samples from the signature-based approaches and behavioral-based approaches. After collecting the samples we combine both features using some feature extraction techniques. In this work total of 3293 features are used for evaluating the hybrid model.

(ii) Feature extraction: Feature engineering is the most important phase of any machine learning technique .feature engineering is the process of selecting and transforming the variable into useful features from the raw data by using some techniques. Typically malware features are extracted by using some data mining tools, such as the n-gram model and graph-based model to create an effective malware dataset and feature. To construct features, the n-gram can employ both static and dynamic properties. n-gram groups system calls or application programming interfaces (APIs) in a

sequential sequence by defined n (n = 2, n = 3, n = 4, n = 6, etc.) variables to construct features from behaviors. Although the n-gram model is commonly employed in malware detection, it has several limitations when selecting characteristics. This is due to the fact that all consecutive static and dynamic properties are unrelated to one another. This makes later steps like classification and clustering more difficult.In order to extract characteristics from these analyzed header components, we created a module by using Python's pefile library. The MS-DOS stub header takes up the first few hundred bytes of a typical PE file. The file header follows the MS-DOS stub and contains abstract information about the whole file. The significant elements of the structure type component IMAGE FILE HEADER in the file header are NumberOfSections, SizeOfOptionalHeader, and Characteristics. Values from these crucial fields are retrieved as features in the proposed study. The NumberOfSection parameter indicates the number of sections shown in the section header. Every executable file must include an optional header. It includes information on how binaries are loaded, such as the size of the data segment and the amount of stack to reserve, among other things. The relevant information is contained in the fields of the component IMAGE OPTIONAL HEADER. These fields in Windows OS include additional information necessary by the linker and loaded. The section header follows the optional header in the PE file header. V1 to Vn are the values from all VirtualAddress fields in the component of type structure IMAGE SECTION HEADER (where n is the total number of sections in the section header). The number of sections field in the file header is exploited in this model to compute the needed value as a feature.

(iii) Standardization: One of the most significant data preparation steps in machine learning is feature scaling. If the data is not scaled, algorithms that compute the distance between the features are biased toward numerically greater values. Tree-based algorithms are somewhat insensitive to feature size. Furthermore, feature scaling aids machine learning and deep learning algorithms in training and convergent learning. The most often used feature scaling strategies are normalization and standardization.

(iv) Evaluation of analysis approaches: The model's performance and experimental results are evaluated using measures such as true positive rate (TPR), false-positive rate (FPR), accuracy, precision, and recall. In the case of a malware detection issue: The number of benign executable files identified as benign is denoted by TN, the number of malicious executable files classed as malicious is denoted by TP, and the number of malicious executable files misclassified as benign is denoted by FN. The number of benign executable files that are incorrectly labeled as malicious is referred to as FP. TPR, FPR, Accuracy, Precision, and Recall are determined as stated in equations.

TPR (True Positive Rate) is calculated by dividing the number of true positives by the total number of malicious executable files.

$$\mathrm{TPR} = \frac{TP}{TP + FN}$$

FPR (False Positive Rate) is calculated by dividing the number of false positives by the total number of benign executable files.

$$FPR = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}$$

Precision is calculated by the sum of true positive and false positive numbers divided by the number of true positives.

$$\text{Precision} = \frac{TP}{TP+FN}$$

The recall is calculated by the True positive numbers divided by the total of true positive and false positive numbers equals recall.

$$\text{Recall} = \frac{TP}{TP + FP}$$

Accuracy is a classification rate that is defined as the sum of the true positive and true negative values divided by the total number of cases.

$$\text{Accuracy} = \frac{TP+TN}{TP+FN+FP+TN}$$

(v) Executable File Classification: The many supervised machine learning classification algorithms are used to identify generalizations and patterns in data that already have class labels. On the gathered notable characteristics, classification methods such as k-Nearest Neighbors, Ada Boost, Random Forest, Bernoulli, and Support Vector Machine are used. Classification methods are employed in this case to categorize the data by training the model and then adding fresh data to the trained model for prediction. With classifiers as a supervisor or learning technique, the model learns to train itself to uncover certain patterns and inferences for prediction. Following training, the model is evaluated against testing data to determine the performance accuracy of the learning approach used to train the data. These classifiers are used to build the various models. The random forest classifier is used to build the best model. Random forest is regarded as the greatest classifier since it comprises numerous decision trees that offer the best classification result overall. The decision tree, an individual component of a random forest, accomplishes dataset splitting in a tree-like structure by executing a feature test at each node that optimizes certain conditions. The Gini Index is the splitting method used by random forest and decision tree classifiers for splitting criteria.

5. OUTPUT AND FUTURE SCOPE

Overall, all the models demonstrate extremely high accuracy in detecting malware, with most achieving near-perfect or perfect accuracy. This suggests that the models are highly effective in distinguishing between malware and non-malware instances in the dataset.

The provided classification evaluation metrics represent the performance of various machine learning models on a dataset. Here's a breakdown:

1.Logistic Regression: Achieves an accuracy of 65%. It has moderate precision, recall, and f1-score for both classes (0 and 1).

2.Decision Tree Classifier: Achieves perfect accuracy (100%) with ideal precision, recall, and f1-score for both classes.

3.KNN (K-Nearest Neighbors): Also achieves perfect accuracy (100%) with optimal precision, recall, and f1-score for both classes.

4.SVC (Support Vector Classifier): Although it achieves 50% accuracy, it fails to predict any instances of class 1, resulting in low precision, recall, and f1-score for class 1.

5.Random Forest: Like Decision Tree and AdaBoost, it achieves perfect accuracy (100%) with optimal precision, recall, and f1-score for both classes.

In summary, Decision Tree, KNN, and Random Forest perform exceptionally well with perfect accuracy, while Logistic Regression shows moderate performance, and SVC fails to predict class 1 instances.

6. CONCLUSION

This study presents an effective and quick way of malware detection. The suggested model is based on static and dynamic analytic approaches, and it combines a hybrid model. The model learns which category the provided file belongs to and whether it is malicious or benign by using machine-learning approaches. The file header, optional header, and section header of different executable files are used to extract features from portable executable files. Extracted characteristics from portable executable files are employed as input to multiple classifiers to diagnose the malware, and the random forest classifier attained the greatest static analysis accuracy of 98% and the highest dynamic analysis accuracy of 94 %. The hybrid model was also trained and evaluated on combined extracted features of file, optional, and section header and reached the best accuracy of 98 % using a random forest classifier. It is found that the accuracy obtained by combining dynamic and static analysis is comparable to the accuracy obtained by hybrid analysis.

REFERENCES:

[1] Samarth Tyagi et.al , AchintyaBaghela et.al, Kashif Majid Dar et.al, Anwesh Patel et.al, Sonali Kothari et.al, SnehalBhosale et.al;IEEE;2023; Malware Detection in PE files using Machine Learning

[2] Muhammad Shoaib Akhtar et.al and Tao Feng et.al; MDPI;2022; Malware Analysis and Detection Using Machine Learning Algorithms

[3] QasemAbuAl-Haijaet.al ,AmmarOdeh et.al and HazemQattous et.al; MDPI;2022; PDF Malware Detection Based on Optimizable Decision Trees

[4] ShouqAlnemariet.aland Majid Alshammari et.al; 2023;MDPI;Detecting Phishing Domains Using Machine Learning

[5] Osama Khalid et.al ,SubhanUllahMudassarAslam et.al, Tahir Ahmad et.al, AttaullahBuriro et.al , Saqib Saeed et.al and Rizwan Ahmad et.al;2023;MDPI; An Insight into the Machine-Learning-Based Fileless Malware Detection

[6] AttaullahBuriro et.al, Abdul BaseerBuriro et.al , Tahir Ahmad et.al , SaifullahBuriro et.al and SubhanUllah et.al ; 2023;MDPI; MALWD&C:A Quick and Accurate Machine Learning-Based Approach for Malware Detection and Categorization

[7] Ali Hussein et.al and Ali Chehab et.al;2023;MDPI; Leveraging Adversarial Samples for Enhanced Classification of Malicious and Evasive PDF Files FouadTrad

[8] Bilal Khan et.al,Muhammad Arshad et.al and Sarwar Shah Khan et.al; 2023;Journal of cyber security;Comparative Analysis of Machine Learning Models for PDF Malware Detection: Evaluating Different Training and Testing Criteria,

[9] Nana Kwame Gyamfiet.al,NikolajGoranin et.al, DainiusCeponis et.al and HabilAntanasCenys et.al;2023;MDPI; Automated System-Level Malware Detection Using Machine Learning: A Comprehensive Review

[10] Ms. AradhanaPawale et.al, Prof. Santosh Biradaret.al ; 2022;IRJETS;MALWARE DETECTION USING MACHINE LEARNING