International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 07 | July - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

Novel Hybrid, Feature-Based Malware Detection Model Using Machine-Learning Techniques

M. Divya[1], CH. Karteek Vamsi[2], D. Pravallika[3] , G.Rama Krishna[4]

Department of CSE, Lendi Institute of Engineering & Technology

---

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 Support Vector Machine, AdaBoost, XGBoost, Random Forest, KNearest Neighbors (KNN), and Bernoulli Naive Bayes (BernoulliNB), k-means. 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.

Key Words: Malware detection, KNN, K-means, SVM, AdaBoost, Random Forest, BernoulliNB

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. 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. 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 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. In this work total of 3293 features are used for evaluating the hybrid model.

Table -1:Sample Table format

File Path	File Type	File Size	Obfuscatic	Benign/Ma	Label
C:\Users\S	Executable	2048	Code Obfu	Malicious	1
C:\Docum	Document	512	None	Benign	0
C:\Prograr	DLL	4096	Packer Ob	Malicious	1
...					
File Path	File Type	File Size	Obfuscatic	Benign/Ma	Label
C:\Users\S	Executable	2048	Code Obfu	Malicious	1
C:\Docum	Document	512	None	Benign	0
C:\Prograr	DLL	4096	Packer Ob	Malicious	1
C:\Downlc	Script	256	Encryption	Malicious	1
C:\System	System	1024	None	Benign	0
C:\Prograr	Executable	10240	Polymorph	Benign	0
C:\Files\in	PDF	768	None	Benign	0
C:\Users\S	Executable	3072	Trojan Hor	Malicious	1
C:\Temp\r	Document	1280	Macro Obf	Malicious	1
C:\Downlc	Executable	6144	None	Benign	0

(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. 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.

(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.

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

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

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

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. 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. 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

SYSTEM ARCHITECTURE

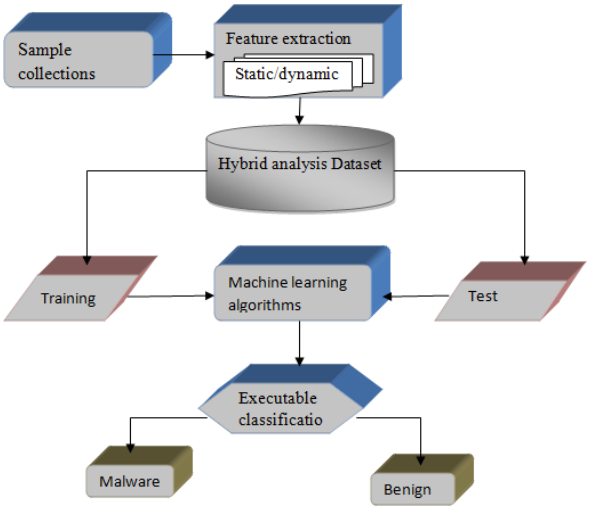


Fig -1

3. ANALYSIS

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.

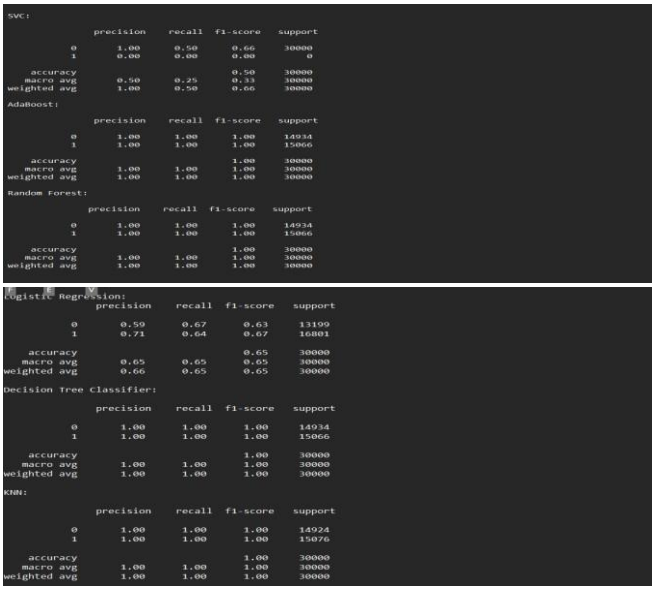
3. KNN (K-Nearest Neighbors): Also achieves perfect accuracy (100%) with optimal precision, recall, and f1-score for both classes.

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. AdaBoost: Similar to Decision Tree and KNN, it achieves perfect accuracy (100%) with ideal precision, recall, and f1-score for both classes.

6. 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, AdaBoost, and Random Forest perform exceptionally well with perfect accuracy, while Logistic Regression shows moderate performance, and SVC fails to predict class 1 instances.



SVC:

	precision	recall	f1-score	support
0	1.00	0.50	0.66	30000
1	0.00	0.00	0.00	0
accuracy			0.50	30000
macro avg	0.50	0.25	0.33	30000
weighted avg	1.00	0.50	0.66	30000

AdaBoost:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	14934
1	1.00	1.00	1.00	15066
accuracy			1.00	30000
macro avg	1.00	1.00	1.00	30000
weighted avg	1.00	1.00	1.00	30000

Random Forest:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	14934
1	1.00	1.00	1.00	15066
accuracy			1.00	30000
macro avg	1.00	1.00	1.00	30000
weighted avg	1.00	1.00	1.00	30000

Logistic Regression:

	precision	recall	f1-score	support
0	0.59	0.67	0.63	13199
1	0.71	0.64	0.67	16801
accuracy			0.65	30000
macro avg	0.65	0.65	0.65	30000
weighted avg	0.66	0.65	0.65	30000

Decision Tree Classifier:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	14934
1	1.00	1.00	1.00	15066
accuracy			1.00	30000
macro avg	1.00	1.00	1.00	30000
weighted avg	1.00	1.00	1.00	30000

KNN:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	14924
1	1.00	1.00	1.00	15076
accuracy			1.00	30000
macro avg	1.00	1.00	1.00	30000
weighted avg	1.00	1.00	1.00	30000

4. CONCLUSIONS

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. [1] Samarth Tyagi et.al , Achintya Baghela et.al, Kashif Majid Dar et.al, Anwesh Patel et.al, Sonali Kothari et.al, Snehal Bhosale 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] Qasem AbuAl-Haija et.al , Ammar Odeh et.al and HazemQattous et.al; MDPI;2022; PDF Malware Detection Based on Optimizable Decision Trees

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

[5] Osama Khalid et.al ,Subhan Ullah Mudassar Aslam et.al, Tahir Ahmad et.al, Attaullah Buriro et.al , Saqib Saeed et.al and Rizwan Ahmad et.al;2023;MDPI; An Insight into the Machine-Learning-Based Fileless Malware Detection

[6] Attaullah Buriro et.al, Abdul Baseer Buriro et.al , Tahir Ahmad et.al , Saifullah Buriro et.al and Subhan Ullah 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 Fouad Trad

[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 Gyamfi et.al,Nikolaj Goranin et.al, Dainius Ceponis et.al and Habil Antanas Cenys et.al;2023;MDPI; Automated System-Level Malware Detection Using Machine Learning: A Comprehensive Review

[10] Ms. Aradhana Pawale et.al, Prof. Santosh Biradar et.al ; 2022;IRJETS;MALWARE DETECTION USING MACHINE LEARNING