Malware Prediction Classifier using Random Forest Algorithm

Hasmitha Dasari[1], Balarka Pradhyumna Danduboina[2], Dr. M. Chinna Rao[3]
[1,2]*Department of Computer Science and Engineering, Lingayas Institute of Management and Technology Vijayawada, Andhra Pradesh, India*
[3]*Professor, Department of Computer Science and Engineering, Lingayas Institute of Management and Technology, Vijayawada, Andhra Pradesh, India*

***Abstract*— Windows devices are also becoming more popular and are more defenseless to malware attacks. Malware is computer code that is designed to harm the operating system and has various names, including adware, spyware, viruses, worms, trojans, backdoors, ransomware and command and control (C&C) bots, depending on its function. Malware attacks on systems are increasing as a result of increased internet use. The detection of unknown malware has been attempted using several strategies, but none of them have been successful. To deal with these threats, proposed research utilized dynamic malware research based on machine learning. Many malicious software-scanning tools are available for Windows PCs, but they perform static analysis, which consumes a lot of time and resources. To address a solution to this problem, an imaging technique to detect malware effectively by converting malware binaries into .exe files and applying machine learning to those .exe files. A comparison of different Windows PC malware detection techniques with different machine learning classifiers is undertaken to detect reliably.**

Index Terms: **Malware attacks, Windows, Machine Learning, Random Forest Algorithm, vulnerable.**

I.INTRODUCTION

Computer systems play an important part in the day-to-day operations of both government and private sector companies. Because computers have become such an important part of any business, it is a big challenge to keep them safe from malicious actions that target the computer's structures and information. Malware is a big issue in modern innovation; it is a type of software that is specifically designed to allow them to access a system without the authorization of the system user, steal data, erase files, and crash the system. Malware detection is an important factor to consider when it comes to computer security. The Malware Detection System is employed in many kinds of situations. Although several techniques have been developed to detect malware in its early stages of development, they have yet to detect malware cases. In the proposed work, a novel dynamic malware method is used since it has multiple execution paths and can cause destructive behavior. Malware analysis is a method for studying malicious activities and determining how to analyze malware's components and behavior. In this paper, the dynamic malware analysis approach was utilized because static analysis is a way of malware investigation that does not involve executing malware, whereas dynamic analysis is a form of malware investigation that involves running malware in a secure manner. Malware can be better understood through behavior analysis. The Feature Extraction technique takes into account [1]. Malware classification based on particular common patterns [1]. Malware families, according to the scientists, exhibit common behavioral traits that indicate their origin and goal. Their approach is divided into three stages: (a) monitoring the behavior of gathered malware in a sandbox environment,

(b) based on a corpus of malware labelled by an anti-virus scanner, and (c) based on a corpus of malware labelled by an anti-virus scanner. (d) Discriminative elements of the behavior models are rated for explanation of classification judgments after a malware behavior classifier is trained utilizing learning approaches.

Malware Instruction Set is a novel representation for harmful software behavior that is being monitored (MIST) [2]. To improve the representation for effective and efficient behavior analysis, data mining

and machine learning technologies are applied. It may be collected automatically during malware analysis by utilizing a behavior monitoring programmed or by converting existing behavior reports. The Random Forest classifier achieves the best results in terms of accuracy, false positive rate, and true positive rate.

The Used machine learning methods to construct a framework for malware de-obfuscation and analysis that performed well in detecting potential risks[3].

There is a survey in that gives a global perspective of how machine learning algorithms can be employed in offense-defense and, more broadly, cyber security[4].

II. LITERATURE REVIEW

To classify malware based on specific shared patterns. According to the authors, malware families have common behavioral patterns that suggest their origin and intent. Their method consists of three stages:

(a) Monitoring the behavior of gathered malware in a sandbox environment.

(b) Based on pre defined pattern of different types of malwares used in today's environment.

(c) Based on a corpus of malware labelled by an anti-virus scanner.

(d) Discriminative elements of the behavior models are rated for explanation of classification judgments after a malware behavior classifier is trained utilizing learning approaches.

Malware Instruction Set is a novel representation for harmful software behavior that is being monitored (MIST). Data mining and machine learning approaches are used to optimize the representation for effective and efficient behavior analysis. It can be obtained automatically using a behavior monitoring tool during malware investigation or by converting existing behavior reports. The Random Forest classifier achieves the best results in terms of accuracy, false positive rate, and true positive rate. There is a survey in that gives a global perspective of how machine learning algorithms can be employed in offense-defense and, more broadly, cyber security. According to the research and observations, machine learning algorithms can outperform traditional malware detection methods used by anti-virus software detection. We've also discussed about several topics algorithms for machine learning that can be very useful in detecting malware. Finally, it can be concluded that this study has produced a proof-of-concept for an alternate malware detection approach. This study used the best algorithm to show feature selection.

III. DATASET COLLECTION

A major part of solving any problem with machine learning is gaining proper dataset for the training model. Getting the proper data consists of gathering or identifying the data that correlates with the outcomes the system wants to predict. In order to find the pattern of malwares, we have to study the behavior of malicious exe files in the real world.

Acquiring the dataset is the first step in machine learning. To build and develop Machine Learning models, you must first acquire the relevant dataset. This dataset will be comprised of data gathered from multiple and disparate sources which are then combined in a proper format to form a dataset. Dataset formats differ according to use cases. For instance, a business dataset will be entirely different from a medical dataset. While a business dataset will contain relevant industry and business data, a medical dataset will include healthcare-related data. You can also create a dataset by collecting data via different Python APIs. Once the dataset is ready, you must put it in CSV, or HTML, or XLSX file formats.

In this section, we describe the proposed system architecture of malware prediction with the help of Figure 1.

A. Collection of Dataset

The first part is the collection of datasets. This dataset will be made up of data collected from various and different sources, which will then be integrated in the right way to produce a dataset.

B. Data Preprocessing

The adjustments we apply to our data before feeding it to the algorithm are referred to as pre-processing. Data preprocessing is a technique for converting raw data into a clean data collection. In other words, anytime data is acquired from various sources, it is obtained in raw format, which makes analysis impossible.

C. Prepare the Data

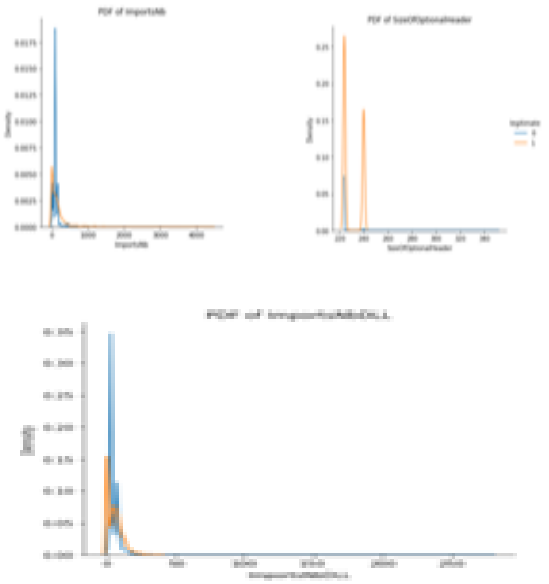
Data preparation is an exploratory data analysis and visualization it divides into statical modelling and claimed to that of no obvious errors. Data pre-processing is the process of transforming the raw data into an understandable format. Here we are pre-processing the data by these ways:

- One-hot categorical data.
- Applying standard Scaler for numerical.

Figure 1: System Architecture

D. Data Visualization

Data visualization is defined as a graphical representation that contains the information and the data. By using visual elements like charts, graphs, and maps, data visualization techniques provide an accessible way to see and understand trends, outliers, and patterns in data. Data visualization provides an important suite of tools for identifying a qualitative understanding. This can be helpful when we try to explore the dataset and extract some information to know about a dataset and can help with identifying patterns, corrupt data, outliers, and much more.



E. Dataset Splitting

Data splitting is when data is divided into two or more subsets. Typically, with a two-part split, one part is used to evaluate or test the data and the other to train the model. Data splitting is an important aspect of data science, particularly for creating models based on data. This technique helps ensure the creation of data models and processes that use data models -- such as machine learning are accurate. In this project we divide into training -70% and for test-30%.

F. Modelling

The process of modeling means training a machine learning algorithm to predict the labels from the features, tuning it for the business need, and validating it on holdout data. A machine learning model is built by learning and generalizing from training data, then applying that acquired knowledge to new data it has never seen before to make predictions and fulfil its purpose. We applied the logistic regression, ADA-boost classification,

Random Forest classifier, Decision trees, KNN algorithm to test the accurate model.

$Hp\ (q) = -\ 1\ N\ \sum yi.\ \log(p(yi\ )) + (1 - yi\ ).\ \log(1 - p(yi\ ))$

IV. MODULE EVALUATION

In this section we discuss the various modules that are used to detect malware. The following are the modules of the proposed system:

1. Training of Dataset: From the collected dataset the extraction of type of files which contain the parameters like size of optional header, characteristics, Machine, etc. We had taken dataset from the UCI Machine Learning Repository and Kaggle. This generated dataset is used to train the random forest model based on our given parameters and the layers. The data is pre-processed and the files which could not be de-compiled are removed. The malicious file set is used for training the model.

2. Modelling: A machine learning model is built by learning and generalizing from training data. Then, it applies that acquired knowledge to new data it has never seen before. This is done to make predictions and fulfill its purpose. We used random Forest regression, decision tree, logistic regression, Ada boost. Out of 5 the random forest algorithm is best fit for our dataset and gave us highest accuracy. We trained model with random forest algorithm.

Random forest: After extracting the features, the random forest algorithm is used for classification. The name if we break down the word, it consists of 'forest' which consists of a group of decision trees, and the word 'random' comes because we are doing random sampling. Applying this algorithm to a data set, it takes a subset of the data as a training set and clusters the data into groups and subgroups. On connecting the data points to groups and sub-groups we get a structure resembling a tree, called a decision tree. The algorithm then prepares a number of trees, resembling a forest. But each tree is different, as for each split in the tree, the variables are chosen randomly. The remaining data set, apart from the training set is used for predicting the tree in forest which makes best classification of data points and the tree having most predictive power is shown as output. Then, a set of labels is set to determine the type of each app where 1 denotes malware and 0 denotes benign apps. At each node of the decision tree, it splits the training set into two subsets with different labels by minimizing the uncertainty of the class labels.

3. Prediction: After training of dataset and finding permission-based features, a model is prepared. A relatively unknown application is sent as new data for predicting malicious of benign, the parameters of random forests at each node of each decision tree are set and have the capability of classifying malicious files. When any new file is given in as input to the Streamlit, which is a user interface. The input files will be evaluated by the trained machine learning model and the output will be displayed whether the file contains malware or not.

V.RESULT AND DISCUSSION

In this section we discuss the evaluation metrics adopted as well as the results compared with other detection techniques. Many experiments are conducted for Malware detection. The comparison with other machine learning approaches is also discussed along with illustrations. In our evaluation test we have kept malicious files as positive samples and clean files as negative samples, thus, we first provide three types of values.

1) (tp: true positive): The number of malicious files that are correctly identified as malicious files.

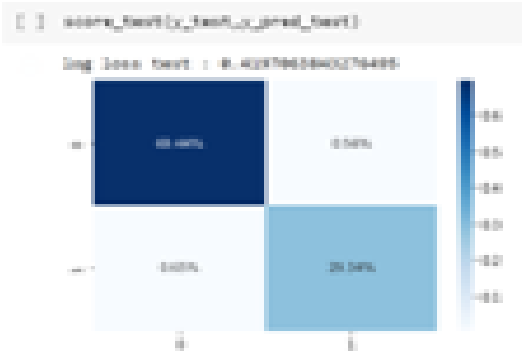
2) (fp: false positive): The number of clean files that are incorrectly identified as malicious files.

3) (fn: false negative): The number of malicious files that are incorrectly identified as clean files.

Therefore, the metrics precision and recall can be calculated as follows: precision = tp/tp + f p recall = tp/tp + f n(5) F 1 = 2 precision recall/precision + recall .

The value of precision falls in the interval of [0, 1], where the large value indicates the correctness of the detection system. The recall equation denotes about how many malicious apps which have been identified by detection system are true malware. The recall value also lies between [0,1].

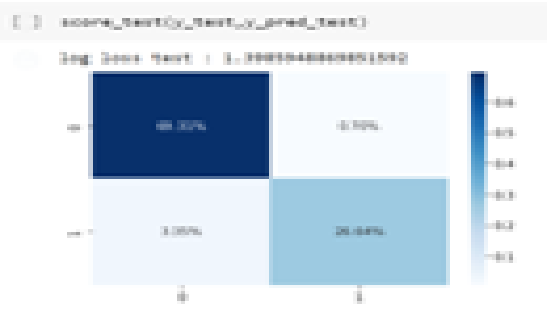
1) ADA Boost: The log loss test of ADA Boost algorithm is 0.4197063849276495.



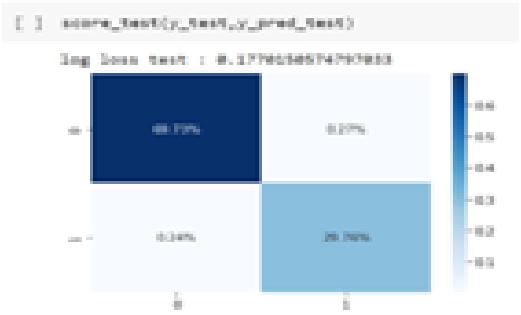
2) Decision Tree: The log loss test of Decision tree algorithm is 0.2952335239494233

3) KNN: The log loss test of KNN algorithm is 0.44410209290437647

4) Logistic Regression: The log loss test of Logistic regression algorithm is 1.3985948869851592



5) Random Forest: The log loss test of random forest algorithm is 0.1770150574797033



Hence, the random forest algorithm has less log loss error and has the ability to predict the difference between malware files and clean files differently.

VI. CONCLUSION

In recent studies and observations, machine learning algorithms have been shown to be more effective than traditional malware detection methods used by anti-virus software. This paper examines the type of file and permissions taken by .exe files. In comparison with KNN, logistic regression, ADA boost, and decision trees, the trained random forest algorithm has less log loss error. Based on the trained random forest algorithm, the test log loss error was predicted at 0.17701. The proposed system takes a relatively short time to execute compared to the existing model. A small dataset of malicious files is used to test this system. The overfitting problem can be avoided in future studies by including a larger dataset. In future work, we will focus on improving the capability of the model to fight against malware efficiently.

VII.ACKNOWLEDGMENT

Our project on dynamic malware detection using machine learning algorithms, we would like to thank our project guide Dr. M. Chinna Rao for helping us throughout our project and giving as important time to time to execute our project.

REFERENCE

[1] Quinlan J R. "Induction of decision tree", Machine Learning, 1: 81~106.
[2] R. Agrawal, "K-Nearest Neighbor Classification for unidentified Data", Journal of Computer-Applications (0975–8887), vol. 105, no. 11, pp. 13-16, 2014.

[3] F. Wei, Y. Li, S. Roy, X. Ou, W. Zhou, "Deep ground truth analysis of current Android malware", Proceeding International Conference Detection Intrusions Malware Vulnerability Assessment, pp. 252-276, 2017.

[4] N.Garleanu, L. Pedersen, "Dynamic Trading, of the foreseeable Returns and the Transaction Costs," Journal of Finance, 2013, vol. 68, issue 6, pp. 2309- 2340.

[5] H. Huang, Z. Wang and W. Chung, "Efficient parameter selection for Support Vector Machine: Business intelligence categorization," 2017 IEEE International Conference on the Intelligence and the Security Informatics (ISI), Beijing, 2017, pp. 158-160, doi: 10.1109/ISI.2017.8004897.

[6] Raff, Zak, Cox, Sylvester, Yacci, Ward. and Nicholas, C., An inspection of byte n-gram characteristics for malware classification, Journals of the Computer Virology and the Hacking Techniques, Volume 14(1), pp. 1-20, September 2016, 2018.

[7] Yu, B., Fang, Y., Yang, Q., Tang, Y., and Liu, L. et al, Analyzation of malware behavior description. Partitions of Information Technology and Electronic Engineering, Volume 19(5), 583-603, 2018.

[8] Gandotra, E., Bansal, D., and Sofat, S, Malware Analysis and Classification: A Survey. Journal of Information Security, Volume 5, pp. 56-64. doi: 10.4236/jis.2014.52006.

[9] Ye, Y., Li, T., Adjeroh, D. and Iyengar, S. S., A Malware Detection Using Data Mining Survey Techniques, Journal ACM Computing Surveys (CSUR), 50(3), pp. 41, Oct 2017.