Machine Learning Techniques for Malware Detection

Prati Jain
Student, dept. of CSE
Faculty of Engineering and Technology,
Jain University
Bangalore, India
pratijain001@gmail.com

Ishita Rajvaidya
Student, dept. of CSE
Faculty of Engineering and Technology,
Jain University
Bangalore, India
ishita2210rajvaidya@gmail.com

Keshav Kumar Sah
Student, dept. of CSE
Faculty of Engineering and Technology,
Jain University
Bangalore, India
keshavsah02@gmail.com

M.K Jayanthi Kannan
Assistant Professor, dept. of CSE
Faculty of Engineering and Technology,
Jain University
Bangalore, India
k.jayanthi@jainuniversity.ac.in

Abstract*—Malware is a term used to describe the types of malicious software that can be used to infect a single computer or the network of an entire company. Viruses and malware are among the most serious risks to online safety that exist right now. There is a serious threat to world security since the volume of malware is increasing at an alarming rate. In order to prevent detection by any antivirus software, all current malware applications tend to incorporate several polymorphic layers or side mechanisms that automatically update themselves at short intervals so that they can remain undetected for longer periods of time. For the identification of malware, we present a flexible framework that allows the use of various machine learning methods, such as decision trees, random forests, and so on. The system's detection rate is greatly improved by selecting the algorithm with the highest level of accuracy. False positive and false negative rates are calculated using the confusion matrix in order to evaluate the system' s overall performance.

Keywords*—Malware, Malware Detection, Machine Learning, Malware Analysis

I. INTRODUCTION

One of the things that people have that is one of the most valuable assets in the modern world is their data and information. Due to the fact that they are so vital to people, ensuring their safety and protecting them at all costs is an absolute must. Malware is essentially software that is created with the intention of wreaking havoc on a computer system, server, or any other type of network. As a result, it can be installed through a wide variety of channels, including phishing emails, any form of infected file, infected websites, and so on.

Therefore, in order to maintain the safety and security of our system, we need to delete all of the files that contain malware, which makes the identification of malware a pressing necessity at this point in time. The detection of malware can therefore assist in the safety of a great deal of sensitive information stored on people's computers and also strengthen the integrity of the systems themselves.

Antivirus software has a track record of being able to identify potentially harmful data found within a system; however, adding malware detection with machine learning will significantly increase the effectiveness of this feature. If our method is utilised, the accuracy of the standard detection systems provided by antivirus companies can be enhanced by approximately 5 or 6 percent. These detection systems currently have an accuracy of approximately 90 percent.

Therefore, the primary purpose of our project is to do a scan on the file that has been provided and determine whether or not it contains any kind of dangerous content. As a result, the primary focus of the project is on the identification of any form of dangerous material within the files that are delivered.

II. LITERATURE REVIEW

Many researchers have proposed machine learning algorithms for detecting malware. Association classifiers, support vector machines, decision trees, random forests, and Naive Bayes are some of the machine learning methods utilised in malware classification. In this part, we offer a few examples of such procedures.

According to Sanjay Sharma, C. Rama Krishna, and Sanjay K. Sahay [1,] in the modern digital age, we deal with a variety of anti-malware solutions for the detection of malware. These anti-malware solutions, however, are dependent on signatures, which are ineffective when it comes to detecting advanced unknown malware, specifically metamorphic malware.
In order to identify malicious software, researchers have utilised the algorithms and methods of machine learning to conduct research on the opcode frequency. The fact that the algorithms used to detect malware achieved the highest levels of accuracy led to the selection of five classifiers for a more in-depth study. These classifiers included algorithms and

approaches like as LMT, NBT, J48 Graft, Random Forest, and Random Tree, amongst others.

According to [3], malware is essentially code developed by an attacker with the intent of harming the user's systems. Examples of malware include backdoors, viruses, Rootkits, and ransomware.

In the past year, around 3,500,000 new types of malware have been found. They plan to present all previously published papers and existing research in the field of malware detection using machine learning.

The D parameter determines the system's level of strictness for classification as Benign or Malware. There were four distinct N values: 2, 4, 6, and 8. The detection ratio was 74.37 percent when the values of N, K, and D were all set to 17.

Chavan and Zende [8] devised a method to detect spyware using data mining and supervised machine learning. They extracted n-gram characteristics from file samples and employed supervised learning techniques to determine whether a file included spyware. Several supervised learning algorithms and different sized N-grams (focused on n = 5) were compared to develop a solution superior to anti-virus software.

Pradosh Subramanyan, Zhixing Xu, Sharad Malik, and others [2] developed an alternative malware scenario in which one model is utilised for each programme, allowing normal executions to be recognised from malicious executions. This methodology is utilised to differentiate between legal and malware-infected executions. Utilized algorithms include logistic regression, support vector machine (SVM), and random forest. The size of the histogram bins must be selected with care.

Using data mining and supervised machine learning approaches, Chavan and Zende [8] devised a method for detecting malware. They employed supervised learning techniques to determine whether or not a file included spyware by extracting n-gram features from file samples. To find a better solution than anti-virus software, N-grams of various sizes (centred on n = 5) and numerous supervised learning methods were compared.

III. EXISTING SYSTEM

Malware detection is commonly done with the help of anti-virus software which analyzes every program in the system to known malware. However, it is widely known in the security world that the existing signature-based method to virus identification is no longer adequate. Conventional signature matching-based antivirus solutions fail to detect polymorphic and new previously discovered dangerous executables. Standard anti-malware focuses on signatures and Polymorphic malware escape these traditional detection technologies which makes this model less reliable.

IV. PROPOSED SYSTEM

Our software is fundamentally composed of three primary components: the user interface, the train module, and the malware test module. The user interface is the first component of our system, and the train module is the second.

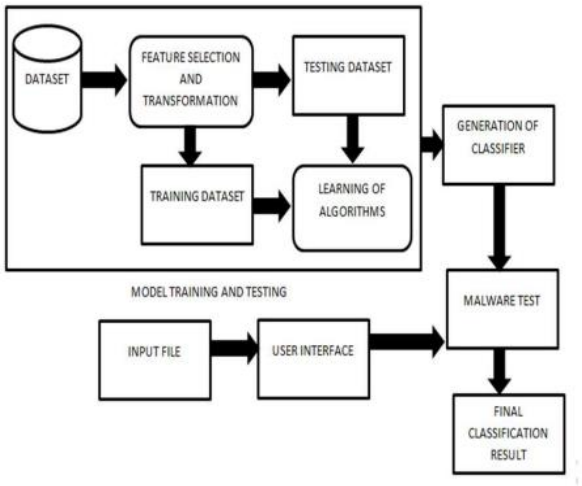
The user interface module is the front-end module, and this module contains the system's front-end architecture. It essentially offers the user with an interface for entering the file to be scanned for dangerous material.

The train module is the subsequent module. This module is used to both train and test the chosen models. The model to be utilised is chosen based on the accuracy of each candidate.

This is the primary module responsible for the final categorization result. In this module, the model's classifier is also generated.

The third module is a test for malware. This module is used to extract data from the file that the user has submitted via the user interface.

It is primarily responsible for the extraction and determination of data from the file, as well as its uploading and division into various sections or characteristics.



The architecture is primarily composed of three modules: 1. Feature Database. 2. Feature Selection and Transformation, and 3. Algorithm Learning

First, we'll go over the dataset. For this project, we used the Kaggle Microsoft malware classification challenge dataset, which is a csv file (comma separated file).

Then, in the following stage, different methods for selecting features are utilised, such as chi-square, information gain, fisher score, gain ratio, and symmetric uncertainty feature selection methods.

Following feature selection and transformation, the dataset will be divided into two parts: Testing Dataset and

Training Dataset. In this case, we employed multiple methods to detect unknown malware in the file.

The architecture's final step is to classify the findings; in the suggested technique, Random Forest, Decision Tree, Linear Regression, and Adaboost detect malware with high accuracy and enhance efficiency.

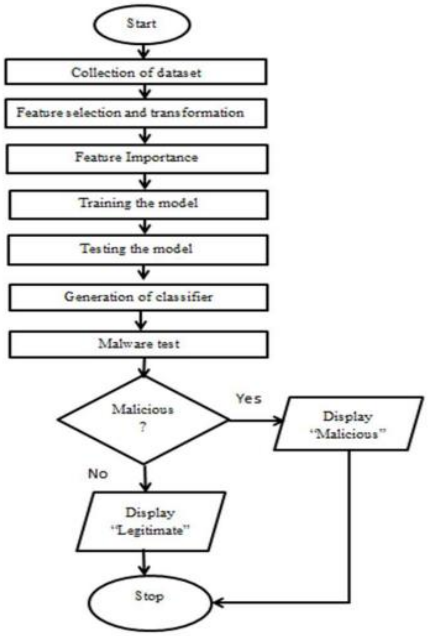
V. IMPLEMENTATION

The project's implementation was carried out using machine learning technologies. Python 3 was used as the programming language for the implementation. Machine learning is the backbone and technology that has been utilised. Flask was used as the front-end technology.

The implementation includes working on three primary project modules: two back-end modules and one front-end module.

Our malware tests and trains the backend modules. The user interface module is the frontend module.

The proper project execution can be explained in a series of steps outlined in a flowchart, which is the process of understanding.



The first step is to collect the data set. This can be accomplished by exploring the web and using websites such as kaggle. After the data is acquired, the data site's features are selected and transformed.

After feature selection and transformation, a crucial step called as feature importance occurs.

The process of determining which traits are the most essential is known as feature importance.

It refers to the features that have the most impact on the database or system. Following that, the data set is divided into two parts:

Training dataset (80% of the dataset):

This component of the dataset is primarily utilised for training. The model basically learns from this dataset.

Testing dataset (20% of the dataset):

This section of the dataset is primarily used for testing. The model is tested using this dataset. The model's accuracy is thus determined using the testing dataset.

```
Now testing algorithms
DecisionTree : 99.040203 %
RandomForest : 99.420500 %
GradientBoosting : 98.851865 %
AdaBoost : 98.641796 %
GNB : 69.916697 %
```

According to our dataset the algorithm with the maximum accuracy is Random Forest.

As a result, it was chosen for usage in the system. Following that, the model is trained on the dataset.

Then two files were generated. They were:

- classifier.pkl
- features.pkl

We choose the testing sample after the classifier is finished. The testing sample is chosen from the testing data that was submitted.The features are then tested with the help of a classifier.

If the file is malicious then the output is displayed as malicious otherwise they output is displayed as legitimate.

VI. CONCLUSION

To maintain the security of our systems, we must ensure that no files include malicious software. Consequently, we designed our technology to detect such malware. During the implementation of the train module, numerous algorithms were used to a dataset in order to get the highest potential accuracy for our system. We choose the most accurate model for a given dataset.
Therefore, based on the findings displayed in the table, it can be concluded that the random forest delivers the highest

accuracy on the dataset. Thus, the random forest classifier is constructed.

This demonstrates that our system's accuracy is approximately 99 percent, which is sufficient for detecting malware. Thus, it can be stated that the results generated by your system have a 99 percent accuracy rate.

A classification approach can be implemented additionally for the presented malware detection system, which will involve the accurate identification of the type of malware that has attacked the file and can serve as a foundation for various research in order to identify the most frequently attacking malwares. Therefore, this provides a suggestion for future project work that can be executed.

REFERENCES

[1] Sanjay K. Sahay, C. Rama Krishna, Sanjay Sharma1,"Detection of Advanced Malware by Machine Learning Techniques:, 2019

[2] Pramod Subramanyan, ZhixingXu, Sayak Ray, Sharad Malik, "Malware Detection using Machine Learning Based Analysis of Virtual Memory Access Patterns",2017

[3] R Mohanasundaram, P Harsha Latha, "Classification of Malware Detection using Machine Learning Algorithms", 2020

[4] Y. K. Penya, Santos, J. Devesa, P. G. Garcia, "N-Grams based file signatures for malware detection", 2009

[5] Thorsten Holz, Konrad Rieck, Carsten Willems, Patrick D¨ussel, Pavel Laskov , "Learning and Classification of Malware Behavior", 2008

[6] J. Zico Kolter, Marcus A. Maloof, Learning to Detect and Classify Malicious Executable in the Wild", 2006

[7] Evgenios Konstantinou, "Metamorphic Virus: Analysis and Detection", 2008

[8] Philip K. Chan, Richard P. Lippmann "Machine Learning for Computer Security", 2006

[9] Hemant Rathore, Swati Agarwal, Sanjay K. Sahay and Mohit Sewak, "Malware Detection using Machine Learning and Deep Learning",2019

[10] Mohd Tanveer Shaikh, Rafia Ansari, Mahenoor Suriya, Sonalii Suryawanshi, "Malware detection using Machine Learning Algorithms", Mohammad Danish Khan, 2017

[11] Jhonattan J. Barriga A. and Sang Guun Yoo, "Malware Detection and Evasion with Machine Learning Techniques: A Survey ", 2017

[12] Priyank Singhal, Nataasha Raul, "Malware Detection Module using Machine Learning Algorithms to Assist in Centralized Security in Enterprise Networks", 2012