IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 9, May 2023

Malware Identification and Classification using Random Forest Algorithms

H Mangalam, Kiruthika K, Pooja J M, Priya T
Department of ECE
Sri Ramakrishna Engineering College, Coimbatore, India

Abstract: *The rapid expansion of malware is now the biggest danger to information security due to the current wave of technical breakthroughs. Numerous thousands of new malware programmes are created daily and propagated around the internet. Malware varieties are always changing, and these harmful software programmes can be categorized as viruses, trojan horses, worms, spyware, botnet malware, ransomware, etc. The identification and categorization of malware is a critical component for many business programmes that provide protection to an organization's data and end-to-end monitoring of the resources accessible by various users. This model can determine a files maliciousness based on its static data or other important characteristics. The idea behind the proposed methodology is to work on the dataset which consists of the signature of malware and identify the characteristics and features to detect the differently classified malwares based on the machine learning algorithms. Our aim is to effectively detect and categorize malware in order to protect the user information from the cyber threats.*

Keywords: Malware, security, detection, classification, maliciousness, cyber threats

I. INTRODUCTION

One of the most significant security risks is malware, which proliferates on its own through human negligence or vulnerabilities. Due to malware's prevalence and destructiveness,it is crucial to identify malware early on when it attempts to infect acomputer or when it begins to assault its hosts. It is necessary to prevent computer infections or to remove malware from a compromised computer system. Malware detection is importantbecause it acts as a computer's early warning system for malwareand cyberattacks, which is important given the prevalence of malware on the Internet. It defends againstdata breaches and prevents hackers from gaining access to the machine.

The term "malware" refers to any program or file that can infecta computer, network, or server and is a contraction for "malicious software". Malware is a generalword for computer programmesthat infiltrate your system and attempt to steal and destroy sensitive data via viruses, worms, Trojan horses, spyware, adware, and othertechniques. This damage may manifest itself to the user or endpoint in a number of ways depending on the form and purpose of the virus. Malware can have terrible effectsin some situations, but it can also have quite moderate and benign ones.

Malware is attractively presented by hackers to persuade victimsto install it. Consumers usually do not realize the program is malware since it seems to be legitimate software.

In essence, that is how malicious software is loaded on a computer. Once it has been installed, malware conceals itself in several computer directories. A virus with sufficient sophistication can get direct access to the operating system. Thenit starts to record private information and encrypt files.

II. LITERATURE REVIEW

Khan, Firoz, et al. [1] proposed that these advocated classifyingthese harmful software programmes as bugs, Trojans, worms, spyware, botnet malware, ransomware, Rootkits, and so on. Many ransomware variants use single or additional strategies intheir assault cycle, as well as Machine Learning systems, to avoid detection. They performed DNA act-Run on 582ransomware besides 942 good ware occurrences to analyse correctness, reminiscence, f-portion, also accuracy to determine the effectiveness of the future technique. The assessment outcomes reveal that, when compared to other methodologies, DNA act-Run can appropriately and efficiently predict and detect ransomware.

Al-Hashmi, Asma A., et al. [2] proposed the identification of malware variants with high accuracy is a critical task. Several current malware variant detection techniques rely on static information derived from the physical erection of the malware folder, such by means of programes and function streams. Static features, on the other hand, may be obfuscated and shelled usingstandard obfuscation techniques. As a consequence, including various behavioural actions into a malware variant detection model can progress recognition exactness though minimising incorrect adverse rates. The researchers created a Deep Collaborative and Multi-layered Behavioural Malware Irregular Recognition Model based on Consecutive Deep Learning and Extreme Gradient Boosting Systems to achieve this aim. Severalbehavioural characteristics were extracted using the dynamic analytic environment. The consequences indicate that the suggested prototypical is consistent, since it advances recognition whereas lowering incorrect rejections.

Xiaofei Xing et al. [3] proposed that this work includes a malware finding model. This model combines a grey-scale image depiction of malware with an autoencoder system, analyses the viability of the grey-scale copy approach of malware constructed on the autoencoder's modernization, mistake, and uses the autoencoder's dimensionality drop features to separate malware after benevolent package in a deep learning model. The proposed detection model outperformed several other well-known machine learning detection methods by means of the Android-side dataset we gathered, with an exactness of 96% and a constant F-score of around 96%.

LinRegDroid et al. [4] proposed a permissions-based malware detection solution for Android. Several different linear regression methods are used in this system. Static analysis is used to obtain application permissions, one of the key components in the safety of the Android functioning structure, and machine learning is used to evaluate the security of the applications. The bagging strategy, one of the ensemble learning techniques, is also used to build a variety of classifiers, which enhances classification performance. Because of this, sorting procedures created on linear regression replicas generate excellent results deprived of necessity for extremely sophisticated arrangement techniques.

Sai, Anne Yeswanth, et al. [5] proposed a flexible architecture that enables the use of a variety of machine learning approaches to differentiate among malware and fresh documents through striving to lower the amount of incorrect positives. In this paper, we first work with cataract one-sided perceptrons, and then with cataract kernelized one-sided perceptrons, to describe the ideas behind our approach. Following successfully testing on medium-sized datasets of malware and fresh documents, the principles underpinning this outline were applied to a grading- up method that allows us to deal by means of actual enormous datasets of malware and fresh documents.

III. METHODOLOGY

Our proposed system is divided into four main modules namely Data Pre-processing, Feature Selection, Malware Detection, Malware Classification. Firstly, in the Data Pre- processing phase dataset consists of characteristics of a malicious files which is gathered from various sources is cleaned by eliminating the unnecessary, missing andredundant data. Wethen proceed with the Feature Selection process where the essential features which is important to detect the malware is identified. We have created a malware detection model to determine if a particular PE file is authentic or malicious. Finally, Malware classification model classifies the malicious fileinto malware families. The product has implemented only bysoftware part. The procedure in software components used in theproduct are explained in detail in this chapter. When random forest is applied to a data set, a portion of the data is used as a training set, and the data is clustered into groups and subgroups.We get a tree-like assembly named a decision tree if we connectthe data points to groups and sub-groups. The programme then arranges the trees in a forest-like pattern. Yet, because thevariables are chosen at random for each split in the tree, each tree is unique. Apart from the training set, the remaining data setis used to forecast the tree in the forest that delivers the best datapoint classification, and the tree with the highest predictive power is shown as output. By applying this algorithm the deep learning-induced detection performance instability is straightforward to evaluate. Using a variety of building pieces such as convolution layers, pooling layers, and fully linked layers. The basis of neural network training is backpropagation.It is a method of fine-tuning neural network weights based on the error rate attained in the previous epoch.

By appropriately selecting the weights, you may reduce error rates and increase the model's dependability by increasing its generalization. To find the best split, each field that might be divided at each node must first be sorted. Finding the appropriate combining weights is crucial since certain algorithms use combinations of fields. Due to the need to construct and evaluateseveral candidate sub-trees, pruning algorithms can also be costly.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 9, May 2023

IV. WORKING PRINCIPLE

To test the model on an unidentified file, it is important to extract the features from the given file. The Python PE file. Fig. 1 demonstrates the usage of the Portable Executable library to create and develop the feature vector as well as the use of an ML model to forecast the class for the supplied file based on the trained model. Our dataset contains the characteristics of the portable executable file (PE File) which is used to predict the given file is malicious (Malware) or a legitimate file. Malware ispackage designed to interfere with, maltreatment, or increase unauthorised that contact to a processer system. Legitimate filesare useful software that don't behave like malware and harmless to the users. The data in the dataset is based on the signatures collected from various malicious software in the past experiences. It holds the technical indicators such as file header data, version information, address of entry point, file size, checksum, DLL characteristics, memory usage and operating system information that can be used to determine whether the given file is malevolent or not. Totally the dataset contains 57 features. The dataset is split into random train and test subgroups using cross validation. The percentage of the dataset to include in the test split is represented by test size = 0.2.

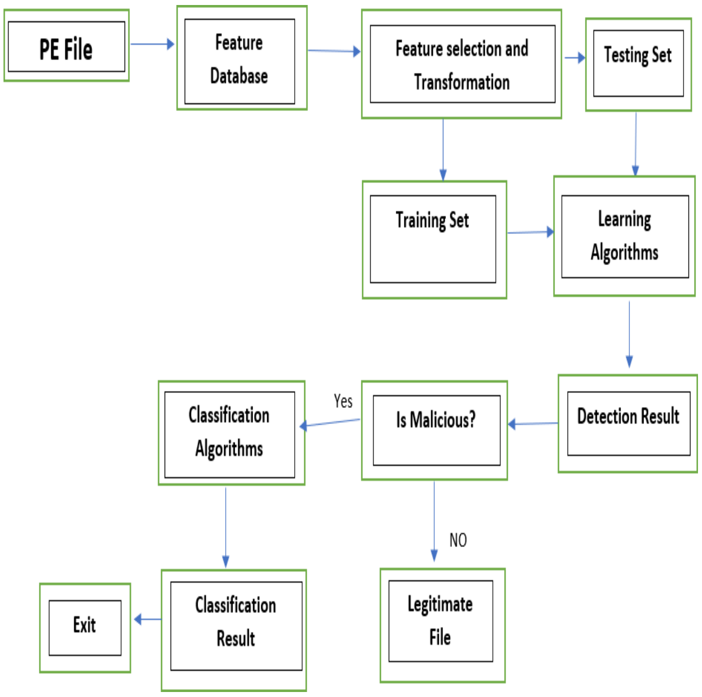


Figure 1 Block Diagram of proposed methodology

Highly Randomized Trees Classifier, often referred to as Extra Trees Classifier, is a category of collaborative learning method that integrates the output of several algorithms of numerous de- interrelated decision trees gathered in a "forest" to get its arrangement consequence. It and a Random Forest Classifier only differ conceptually in terms of in what manner the decision trees in the forest are erected. Each decision tree in the Extra Trees Forest is created by means of the first exercise. Each tree is then specified a random trial of k topographies from the feature-set at each examination node, from which it obligation choose the feature that will best split the data conferring to a accurate standard (stereotypically the gini Guide). As a result of thisrandom sample of features, many de-interrelated decision trees are constructed. In order to carry out feature assortment by means of the above-mentioned foresterection, the overall normalised reduction in the accurate standards used in the split decision feature (gini Guide, if the gini Guide is used in the forest erection) is calculated foreach feature during the forest building. The gini Consequence of the feature is the name given to this value. The process for choosing features involves rating eachfeature rendering to its gini Consequence, then letting the operator choose the highest k characteristics that speak to them.

It is suitable for ML problems including mutually arrangement and regression. It is erected on the impression of collaborative learning, which is a technique of participating several classifiers to tackle complex problemsand increase prototypical performance. Random Forest is aclassifier that, as the name implies, customs a amount of decision trees on distinct subcategories of the input dataset and norms them to enhance the prediction exactness of the dataset. Rather

IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Impact Factor: 7.301

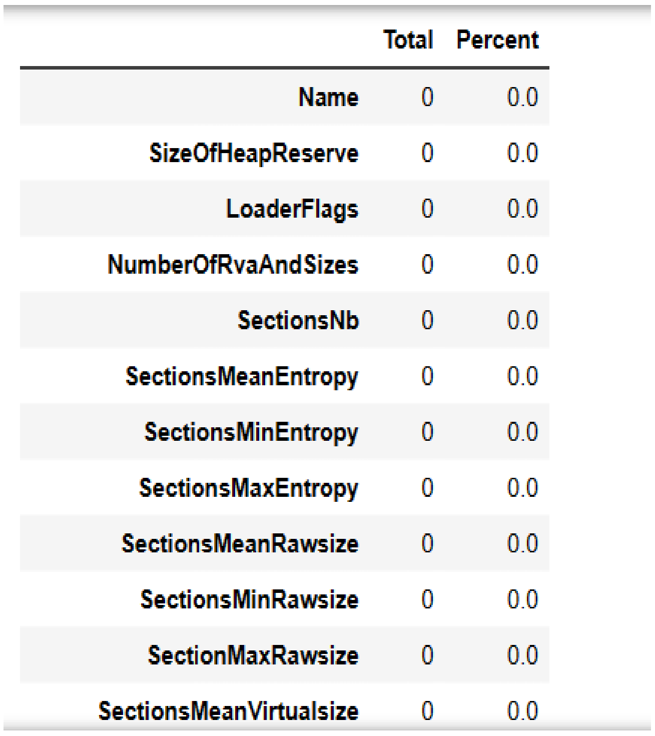
Volume 3, Issue 9, May 2023

of contingent on a solitary decision tree, therandom forest takes into account calculations from each treealso forecasts the future grounded on the mainstream of divisions from projections. The increasing amount of trees in the forest prevents sophisticated precision also overfitting. Approximately decision trees might magnificently do in advance the outcome although others may not since the random forest balances a variety of trees to estimate the dataset's class. Yet, once all of the foliage are connected, they correctly forecast the outcome. The feature variable in the dataset should have some real valuesin order for the classifier to predict correct outcomes ratherthan an assumed conclusion. There must be very littlerelationship between the forecasts from each tree. It takes shorter training time than other algorithms. Despite the massive dataset, it functions efficiently and predicts the conclusion with great accuracy. When a significant amountof the data is absent, accuracy can still be maintained.

V. SOFTWARE IMPLEMENTATION

DATA PREPROCESSING

Data preparation is critical in machine learning constructed replicas, particularly in malware recognition, where misclassification can damage the system. The Fig 2 demonstrates how data pretreatment aids in removing the influence of extraneous material that contributes to categorization in order to optimize accuracy. We can end upwith data that is an erroneous feature depending on our datacollection methods and sources. Unstructured text data make up the majority of the data gathered in the initial phase. It might be dumped in a variety of forms and structures from various acquisition tool types. These types of data frequently contain unnecessary or redundant features, missing values, and noise like symbols, punctuation, and stop words. Such unnecessary information or contradictions ought tobe eliminated since they might lead to false conclusions. In order to aid the machine learning set of rules in finding a proper also illustrative malware outline that can be distinguished from the benign pattern, the data is cleaned by eliminating irrelevant data during the preprocessing step.

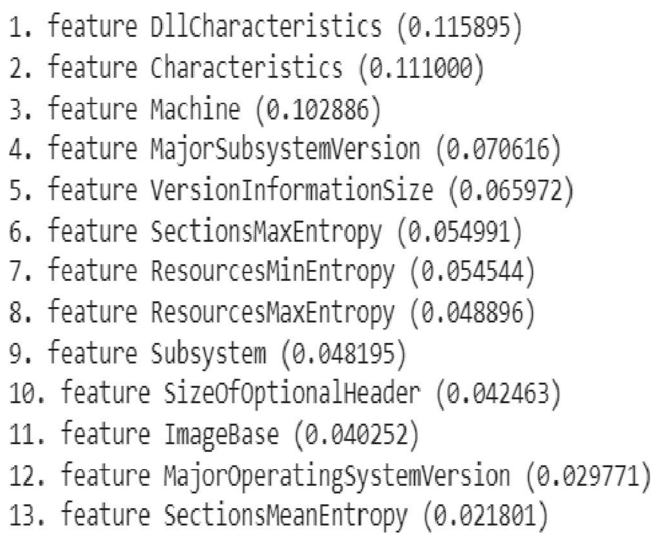


	Total	Percent
Name	0	0.0
SizeOfHeapReserve	0	0.0
LoaderFlags	0	0.0
NumberOfRvaAndSizes	0	0.0
SectionsNb	0	0.0
SectionsMeanEntropy	0	0.0
SectionsMinEntropy	0	0.0
SectionsMaxEntropy	0	0.0
SectionsMeanRawsize	0	0.0
SectionsMinRawsize	0	0.0
SectionMaxRawsize	0	0.0
SectionsMeanVirtualsize	0	0.0

Figure 2 Data Preprocessing

FEATURE SELECTION

By eliminating unnecessary features, feature selection lowers storage and computational costs. The fig. 3 depicts that the huge search area in scenarios with a large number of characteristics, finding the optimal selection of features is a difficult and complex operation. Feature selection is the processof removing the unnecessary or noisy features from the original feature set and picking the most relevant features. Extra Trees Classifier uses be more or less to upsurge prediction exactness and switch over-suitable by suitable a amount of randomized decision trees. To execute feature selection using the aforementioned forest erection, the normalized entire decrease in the accurate standard employed in the feature of divided decision (gini Guide if the gini Guide is used in the forest building) is resolute for individual feature throughout the forestdevelopment. This is recognized as the gini Consequence of thecharacteristic. The feature assortment technique comprises rating to each feature based on its gini Consequence, with the manipulator choosing the highest k topographies that request towards them. In our proposed system 14 features are identifiedas required by Extra Tree Classifier

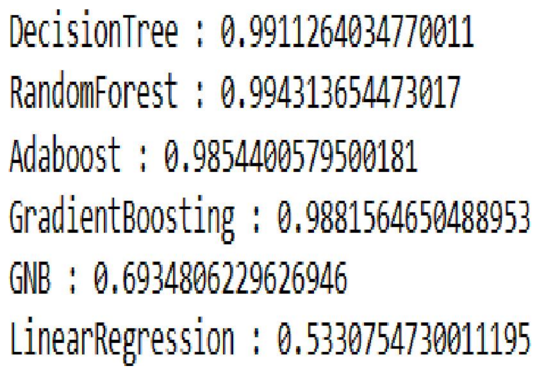


```
1. feature DllCharacteristics (0.115895)
2. feature Characteristics (0.111000)
3. feature Machine (0.102886)
4. feature MajorSubsystemVersion (0.070616)
5. feature VersionInformationSize (0.065972)
6. feature SectionsMaxEntropy (0.054991)
7. feature ResourcesMinEntropy (0.054544)
8. feature ResourcesMaxEntropy (0.048896)
9. feature Subsystem (0.048195)
10. feature SizeOfOptionalHeader (0.042463)
11. feature ImageBase (0.040252)
12. feature MajorOperatingSystemVersion (0.029771)
13. feature SectionsMeanEntropy (0.021801)
```

Figure 3 Selected Features

VI. MODEL COMPARISON

To choose a model, Fig. 4 depicts the model. This method compares the outcomes of six different categorization algorithms before picking which one to utilize for prediction. Linear Regression, Random Forest, Decision Tree, Adaboost, Gaussian, and Gradient Boostingwere some of the machine learning replicas that have been attempted.



```
DecisionTree : 0.9911264034770011
RandomForest : 0.994313654473017
Adaboost : 0.9854400579500181
GradientBoosting : 0.9881564650488953
GNB : 0.6934806229626946
LinearRegression : 0.5330754730011195
```

Figure 4 Model Comparison

LINEAR REGRESSION

Finding the link between two continuous variables maybe done with the use of linearregression. A variable's valuecan be predicted using the value of another variable. The term "dependent variable" describes the variable you'retrying to forecast. "Independent variable" refers to the variable you are using to forecast the value of the other variable.

RANDOM FOREST

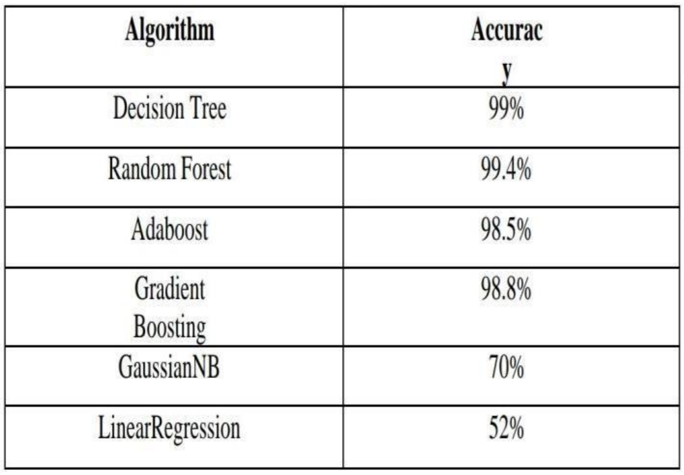
When it comes to solving classification and regression issues, machine learning performances similar to random The abbreviation “Adaboost stands for “Adaptive Boosting,” which is highly well-liked boosting approach that turns several “poor classifier” into one “strong classifier.”

GAUSSIAN

Compared to more well-known machine learning techniques like linear regression models, perceptron-based models, gaussian process models may not be as well recognized. This is problematic because one of the few machine learning methods that can do this is the Gaussian process model still simulate highly complicated systems while still being analytically solvable.

GRADIENT BOOSTING

In gradient boosting, each calculation modifies the error of its preceding. Unlike Adaboost, each predictor is taught by labelling the predecessor's residual errors rather than alteringthe weights of the training samples. The gradient Boosted Trees approach uses CART as its basic learner (Classification and Regression Trees). The model is either deployed in cloudplatforms or saved in local computer. The mainreason behindusing Random forest algorithm is it provides higher accuracythan other traditional machine learning algorithms which wehave tried.



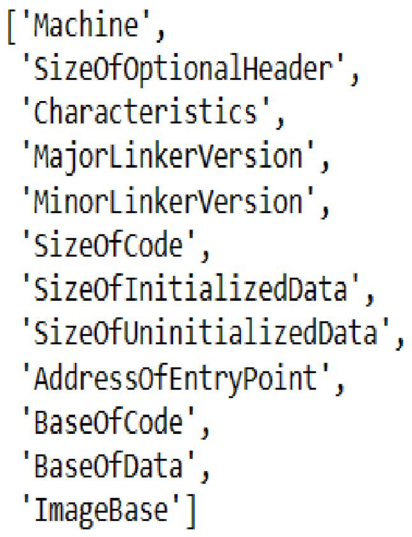
Algorithm	Accurac y
Decision Tree	99%
Random Forest	99.4%
Adaboost	98.5%
Gradient Boosting	98.8%
GaussianNB	70%
LinearRegression	52%

Figure 5 Model Comparison

VII. RESULTS AND DISCUSSION

MALWARE DETECTION:

This section illustrates briefly how a machine learningsystem finds the given file is legitimate or a malicious file. Fig. 6 depicts as a result of model comparison we clearly know that Random Forest algorithm provides higher accuracy among other traditional machine learning algorithms. Random Forest integrates numerous decision trees into a single one for better prediction results. The accuracy rate of Random Forestalgorithm is depending on the number of trees. Higher the number of trees, higher willbe the accuracy rate. Initially, we have loaded the train data to our model for training. After training, the model is dumped into a pickle module. We can serialise and de- serialise a python object structure using the python pickle package. With Python, any object may be pickled and savedto disc. Before delivering an item to a file, Pickle "serializes" it. Pickling converts a Python object (list, dictionaries, etc.) into a character stream. This character stream is expected to provide all of the information needed to recreate the item in a different Python script, a process known as unpickling. A real-time unforeseen PE file (Portable Executable) is given to the model to predict whether it is malware or not. Our proposed model analyses the essential characteristics of the given file and matches it with the dataset contains the historical data about the file characteristics seen before. Finally, the proposed model detects the given file is legitimate or a malicious file with the accuracy of 99.4%.



```
['Machine',
 'SizeOfOptionalHeader',
 'Characteristics',
 'MajorLinkerVersion',
 'MinorLinkerVersion',
 'SizeOfCode',
 'SizeOfInitializedData',
 'SizeOfUninitializedData',
 'AddressOfEntryPoint',
 'BaseOfCode',
 'BaseOfData',
 'ImageBase']
```

Figure 6 Malware Detection

MALWARE CLASSIFICATION:

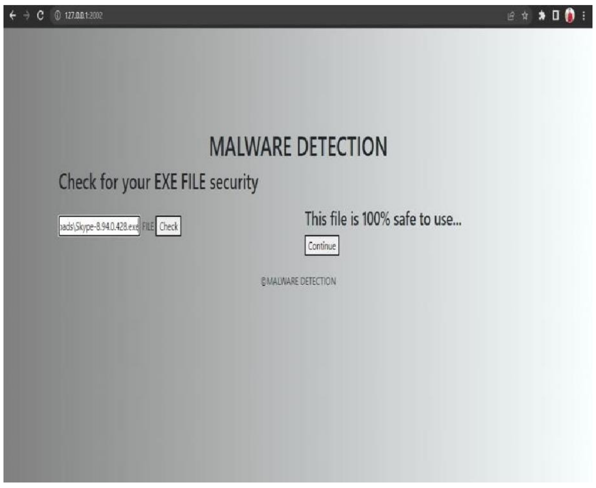
The typical vector acquired in the earlier step is utilized as input characters for the Random Forest method for classificationin this phase. Random Forest model is compared with few othermodels such as CNN, LSTM, Multi-layer perceptron, OVR (One-vs-Rest). The comparison of each of these models by the results they predict for the unlabelled data with the actual labelsof this data is analyzed and Random Forest Classifier seems to work best for this multi-class problem, with an accuracy of 97.14%.

In this machine learning task, 7 malware families are employed for training and prediction. Each is given a numerical value ranging from 0 to 6, as follows:

(BHO, CeeInject, FakeRean, OnLineGames, Renos, Vobfus, and Winwebsec) (0, 1, 2, 3, 4, 5, 6).

FINAL UI RESULT

As a result we have developed the UI page. While weupload the portable executable file path it will check whether the executable file is legitimate or not.



127.0.0.1:2002

MALWARE DETECTION

Check for your EXE FILE security

oads\Skype-8.94.0.428.exe FILE Check

This file is 100% safe to use...

Continue

©MALWARE DETECTION

Figure 7 The file is legitimate

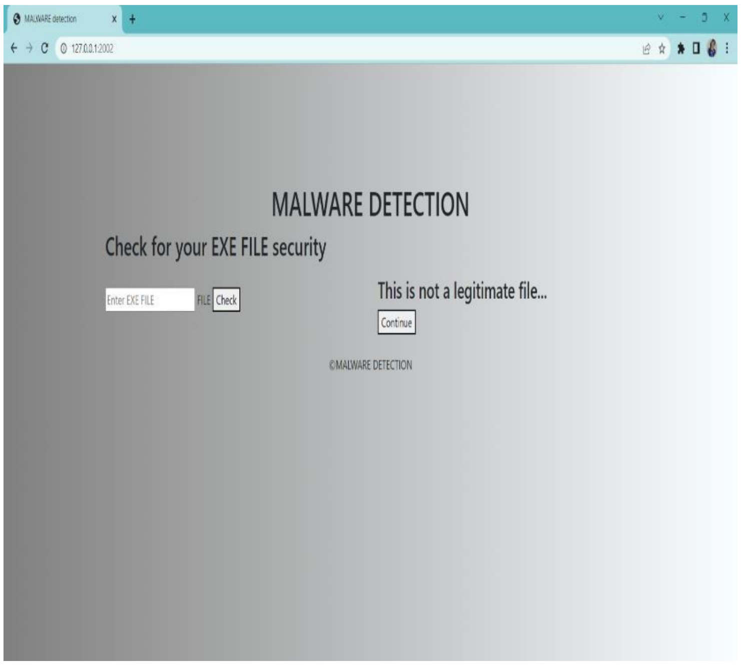


Figure 8 The file is not legitimate

Although there has been extensive study on malware detection and classification, the ability to accurately identify malware variants still poses a severe danger to cyber security. The study demonstrates that the Random Forest model for effectively detecting and classifying the malware. It highlights how important machine learning methods are for organizing and categorizing malware testing for both small datasets and massive amounts of data. One of the essential is process done in this proposed system is Feature Selection. Eliminating features from the feature selection procedure for classification reduces the preparation time for grouping as well. Our proposed model using Random Forest provides 99.4% on detecting the malware and 97.14% on classification of malware. It clearly shows the better performance than other traditional machine learning algorithms.

VIII. FUTURE SCOPE

Future work will involve putting the suggested method to use on a massive dataset and engaging in a thorough analysis to identify malicious software in packed binaries, which is currently thought to be particularly problematic. We will continue to work and explore more methods using deep learning algorithms. A deep learning model may be trained end-to-end to learn complicated feature hierarchies and include many malware detection pipeline steps into a single, robust model that is taught simultaneously. We consider working in the enhanced version of the malware detection with high accuracy and efficiency to evaluate on the real time malware detection and classification.

REFERENCES

[1] Shruti Gedam, Khan, Firoz, Cornelius, Laskshmana kumar Ramasamy. "A digital DNA sequencing engine for ransomware detection using machine learning." IEEE Access 8 (2020): 119710- 119719.

[2] Al-Hashmi, Asma A., Fuad A., Ghaleb, A. Al-Marghilani. "Deep- Ensemble and Multifaceted Behavioral Malware Variant Detection Model." IEEE Access 10 (2022): 42762-42777.

[3] Xing Xiaofei, Xiang Jin, Haroon Elahi, Hai Jiang, Guojun Wang. "A Malware Detection Approach Using Autoencoder in Deep Learning." IEEE Access 10 (2022): 25696-25706.

[4] Şahın, Durmuş Özkan, Sedat Akleylek, and Erdal Kiliç. "LinRegDroid: Detectionof Android malware using multiple linear regression models-based classifiers." IEEE Access 10 (2022): 14246-14259.

[5] Sai, Anne Yeswanth, et al. "Malware Detection Using Machine Learning Techniques." Smart Data Intelligence. Springer, Singapore, 2022. 95-107.

[6] Almomani, Iman, Aala Alkhayer, and Walid El-Shafai. "An Automated Vision- Based Deep Learning Model for Efficient Detection of Android Malware Attacks." IEEE Access 10 (2022): 2700-2720.
[7] Aslan, Ömer, and Abdullah Asim Yilmaz. "A new malware classification framework based on deep learning algorithms." Ieee Access 9 (2021): 87936-87951.
[8] Mahajan, Ginika, Bhavna Saini, and Shivam Anand. "Malware classification using machine learning algorithms and tools." 2019 Second international conferenceon advanced computational and communication paradigms (ICACCP). IEEE, 2019.
[9] Ana, Madhurima, and Swathi Edem. "An Efficient Deep Learning Based Approach for Malware Classification." Machine Learning Technologies and Applications. Springer, Singapore, 2021. 193-201.
[10] Usman, Nighat, et al. "Intelligent dynamic malware detection using machine learning in IP reputation for forensics data analytics." Future Generation Computer Systems 118 (2021): 124- 141.