Malware Detection and Analysis Using Machine Learning

Lakshit Dhanuka
School of Computer Science and Engineering
Vellore Institute of Technology, Tamil Nadu

Nishit Surana
School of Computer Science and Engineering
Vellore Institute of Technology, Tamil Nadu

Rohit Mehta
School of Computer Science and Engineering
Vellore Institute of Technology, Tamil Nadu

Mohd Umar
School of Computer Science and Engineering
Vellore Institute of Technology, Tamil Nadu

Dr. Manikandan K
School of Computer Science and Engineering
Vellore Institute of Technology, Tamil Nadu

***Abstract*—This project aims to present the functionality and accuracy of five different machine learning algorithms to detect whether an executable file is malicious or legitimate. Malware discovery is typically consummated with the assistance of hostile to infection programming which believe about each program in the substructures to known malwares. One. We could utilize the known highlights of malwares and train a model to anticipate if a program is a malware. Along these lines, we will utilize Machine Learning calculations to anticipate if a specific program is a malware or not. Information has been soaring since the appearance of web. Additionally, the kind of information is changing quickly with time. Henceforth, we have to discover devices that could cycle and help in examining various sorts of information effectively and rapidly as the datasets of genuine world have gigantic information storehouses.**

In this task we plan to do as such by utilizing Ember dataset which is an open Dataset for Training Static PE Malware Machine Learning Models. The dataset incorporates highlights separated from 1.1M double records: 900K preparing tests (300K malevolent, 300K favorable, 300K unlabeled) and 200K test tests (100K noxious, 100K kind). To go with the dataset, we likewise discharge open-source code for extricating highlights from extra parallels so extra example highlights can be attached to the dataset. This dataset makes up for a shortcoming in the data security AI people group: an amiable/pernicious dataset that is enormous, open and general enough to cover a few fascinating use cases.

I. INTRODUCTION

Malware is any software purposely designed to cause defilement to a computer, server, client or computer network. This single term circumscribes Viruses, Trojans, Worms etc., whereas Malware Analysis allude to the process of understanding the behavior and purpose of a suspicious file or URL. Machine Learning can be an enchanting equipment for either an essential discovery capacity or analysis of Malware. Machine learning models consequently misuse complex connections between document ascribes in preparing information that are separating among vindictive and generous examples. Besides, appropriately regularized AI models sum up to new examples whose highlights and names follow a comparative dispersion to the preparation information. In any case, it is generally identified in the security network that the current mark- based way to deal with infection identification is not, at this point satisfactory. A simple classification of malware consists of file infectors and stand-alone malware.

The problem to be examined involves the high spreading rate of computer malware (viruses, worms, Trojan horses, rootkits, botnets, backdoors, and other malicious software) and conventional signature matching-based antivirus systems fail to detect polymorphic and new, previously unseen malicious executables. Static executable investigation offers a potential answer for the issues of dynamic examination. Static investigation takes a gander at the structures inside the executable that are fundamental with the end goal for it to run. Since these structures are commanded by the record type, they can't be eliminated, encoded (in spite of the fact that their code substance might be), or muddled without any problem. One proposed approach (solution) is by using automatic dynamic (behavior) malware analysis combined with data mining tasks, such as, machine learning (classification) techniques to achieve effectiveness and efficiency in detecting malware.

A survey on different machine learning methods that were proposed for malware detection is given in Additionally, on the grounds that it just includes parsing structures, it is substantially less computationally costly than dynamic examination. Some exploration has just been done into static executable examination for Windows compact executable (PE) documents. Of specific note, the last 4 or 5 years has seen various progressed tenacious

danger (APT) malware crusades focused on explicitly at Macs. Investigating instruments to resist these dangers currently guarantees that we will have the option to all the more likely handle the expanding danger later on.

II. LITERATURE SURVEY

[1] Analysis of Machine Learning Techniques Used in Behaviour-Based Malware Detection: The problem to be examined involves the high spreading rate of computer malware (viruses, worms, Trojan horses, rootkits, botnets, backdoors, and other malicious software) and conventional signature matching-based antivirus systems fail to detect polymorphic and new, previously unseen malicious executables. The data set consists of malware data set and benign instance data set. Both malware and benign instance data sets are in the format of Windows Portable Executable (PE) file binaries.

[2]Malware Analysis and Detection in Enterprise Systems: The purpose of this research is to investigate techniques that are used in order to effectively perform Malware analysis and detection on enterprise systems to reduce the damage of malware attacks on the operation of organizations. Two techniques of malware analysis which are Dynamic and Static analysis on two different malware samples .The results showed that Dynamic analysis is more effective than Static analysis. Static analysis of malware is defined as the process of extracting information from malware while it is not running by analysing the code of the malware to determine its true intention.

[3] Malware Detection using Machine Learning Based Analysis of Virtual Memory Access Patterns: Malicious software, referred to as malware, continues to grow in sophistication. Past proposals for malware detection have primarily focused on software-based detectors which are vulnerable to being compromised. Thus, recent work has proposed hardware-assisted malware detection. In this paper, they introduce a new framework for hardware-assisted malware detection based on monitoring and classifying memory access patterns using machine learning. This provides for increased automation and coverage through reducing user input on specific malware signatures.

[4]Analysis of features selection and machine learning classifier in android malware detection: The proliferation of Android-based mobile devices and mobile applications in the market has triggered the malware author to make the mobile devices as the next profitable target. With user are now able to use mobile devices for various purposes such as web browsing, ubiquitous services, online banking, social networking, MMS and etc, more credential information is expose to exploitation. Applying a similar security solution that work in Desktop environment to mobile devices may not be proper as mobile devices have a limited storage, memory, CPU and power consumption.

[5] Evaluation on Malware Analysis: In this technological era everyone's life is influenced by Internet. It plays an essential role in today's life style and businesses. Sharing information, communication, socializing, shopping, running businesses and many more are now easily achievable by internet. In spite of its vitality, the Internet experiences significant inconveniences, for example, clients' privacy, robbery, fraud and spamming

III. PROBLEM STATEMENT

Malwares are a prominent form of attack in the Cyber security domain which harms the user data, network, server and so on. Detecting these Malwares is a major challenge as the Malwares authors always come up with unique ways of hiding the source and leveraging the weaknesses of the existing system. We have to identify the parameters and patterns in the PE file which is a challenging task. In this project, we use Machine Learning to detect Malicious file

IV. OBJECTIVE

The scope of this project circumscribes the following objectives: -

1. Identify the parameters that distinguish a

Malicious PE file from a genuine one.

2. Identify patterns in these PE files.

3. Train various classifier models to predict the legitimacy of PE file.

4. Define relevant metrics to gauge the accuracy of the model.

5. Factoring in the statistical records and make predictions on the given data set and identify the

malicious PE file.

V. DATASET

We gathered 51,000 special kind Windows PE records from endpoints and removed highlights from them. We utilized our reaping framework to gather 17,000 one of a kind pernicious Windows PE documents from different sources, these PE were confirmed as noxious utilizing Virus Total. We put away all the malware test documents in a spotless climate and removed from the vindictive PE records similar arrangement of highlights extricated from the kind PE records

VI. ALGORITHM MODELS

1. RANDOM FOREST

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

2. ADABOOST

AdaBoost, short for Adaptive Boosting, is a machine learning that can be used in conjunction with many other types of learning algorithms to improve performance. AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers. AdaBoost is sensitive to noisy data and outliers. In some problems it can be less susceptible to the overfitting problem than other learning algorithms.

3. GRADIENT BOOSTING

Gradient boosting is a machine learning technique for regression and classification problems, which

produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

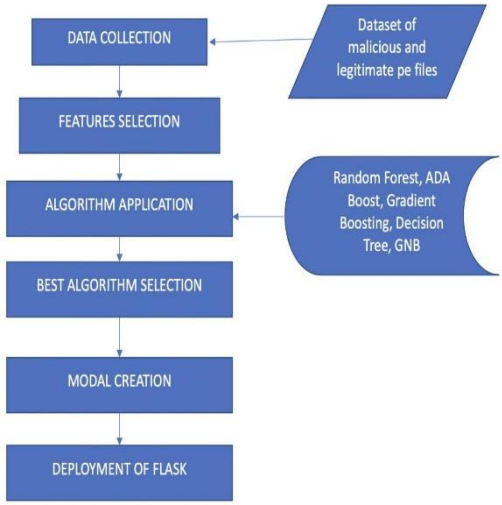
4. DECISION TREE

Decision Tree is an ensemble learning method for classification, regression and other tasks. It works by classifying various instances by sorting the down the tree from the root to some leaf node, which provides the class of the instance. Each node tests the attribute of an instance whereas each branch descending from the node specifies one possible value of this attribute.

5. GNB

Gaussian Naïve Bayes Classifier is a subset of classification algorithms based on Bayes' Theorem. It works on assumption that every pair of features being classified is independent of each other and affects the dependent variable equally. These assumptions are not generally correct in real-world situations but often works well in practice.

VII. PROCEDURES FOR PROPOSED METHOD WITH AN ALGORITHM WORKFLOW DIAGRAM



VIII. ALGORITHM

• Collection of data. The data is taken from VirusShare and has 138047 instances across 54 different attributes.

• As many features are not helpful to differentiate between legitimate and malicious file, the important features are selected using tree classifier.

• A total of 5 supervised classification namely Random Forest, ADA Boost, Gradient Boosting, Decision Tree, GNB are used.

• The best algorithm(Random Forest) is decided on the basis of accuracy of different algorithm.

• Saving the algorithm(classifier.pkl) and feature

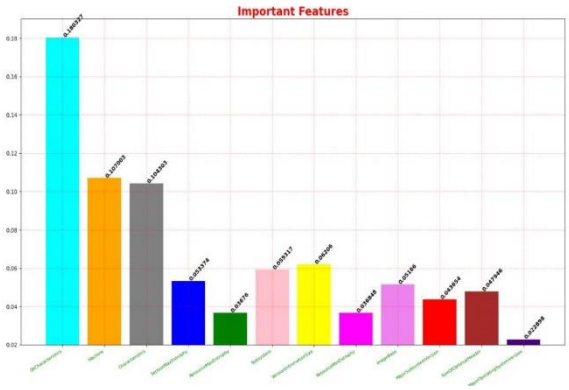
list(features.pkl) for later use.

• Deploying flask web framework.

VIII. RESULTS AND DISCUSSION

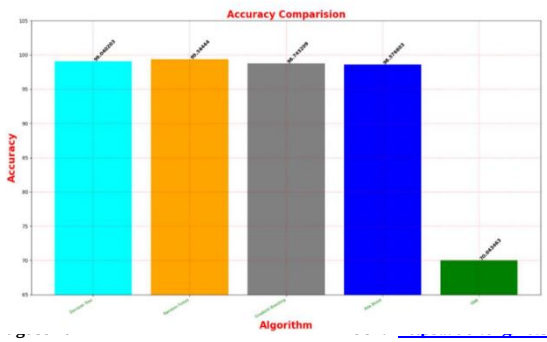
The below table is the shows the Features which affects the machine learning model the most along with their values. It is obtained from learning.py.

Feature Name	Value
Dll Characteristics	0.180327
Machine	0.107003
Characteristics	0.104303
Section Max Entropy	0.053374
Resource Max Entropy	0.036760
Subsystem	0.059317
Version Information Size	0.062060
Resource Min Entropy	0.036848
Image Base	0.051660
Major Subsystem Version	0.043654
Size Of Optional Header	0.047946
Major Operating System Version	0.022898



The below table is the shows the accuracy of each machine learning model used in the project. It is obtained from learning.py.

Algorithm	Accuracy
Decision Tree	99.040203
Random Forest	99.344440
Gradient Boosting	98.743209
ADA Boosting	98.576603
GNB	70.043463



As it can be concluded from the above table that the accuracy of Random Forest is highest, therefore Random Forest is used to test the new PE file to classify them into malicious or legitimate file.

IX. CONCLUSION

The aim of the project is to present the machine learning approach to malware problem has been full filled. The machine learning algorithms applied were Decision Tree, Random Tree, Naïve Bayes, Gradient Boosting and ADA Boosting.

After application it was observed that the Random Forest is the best algorithm for our under taking with an accuracy of 99.344440. This project can reach the application level with the help of library called pickle, to save what the algorithm has learned

X. FUTURE WORK

1. The model can be more accurate and time efficient by adding more data.

2. To host the model on web for real time analysis of exe files on the cloud

3. Increasing the scope of model by including not just static but analysis of Dynamic Malware too.

4. Applying different algorithms to improve the performance.

REFERENCES

[1] Firdausi, Ivan, Alva Erwin, and Anto Satriyo Nugroho. "Analysis of machine learning techniques used in behavior-based malware detection." 2010 second international conference on advances in computing, control, and telecommunication technologies. IEEE, 2010.

[2] Mokoena, Tebogo, and Tranos Zuva. "Malware analysis and detection in enterprise systems." 2017IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC). IEEE, 2017.

[3] Xu, Zhixing, et al. "Malware detection using machine learning based analysis of virtual memory access patterns." Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017. IEEE, 2017.

[4] Mas' ud, Mohd Zaki, et al. "Analysis of features selection and machine learning classifier in android malware detection." 2014 International Conference on Information Science & Applications (ICISA). IEEE, 2014. 36 | P a g e

[5] Agrawal, Monika, et al. "Evaluation on malware analysis." International Journal of Computer Science and Information Technologies 5.3 (2014): 3381- 3383.

[6] Ahmed, Faraz, et al. "Using spatio-temporal information in API calls with machine learning algorithms for malware detection." Proceedings of the 2nd ACM workshop on Security and artificial intelligence. 2009.

[7] Sethi, Kamalakanta, et al. "A novel malware analysis framework for malware detection and classification using machine learning approach." Proceedings of the 19th International Conference on Distributed Computing and Networking. 2018.

[8] Ye, Yanfang, et al. "Combining file content and file relations for cloud based malware detection." Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. 2011.

.

[9] Bearden, Ruth, and Dan Chai-Tien Lo. "Automated microsoft office macro malware detection using machine learning." 2017 IEEE International Conference on Big Data (Big Data). IEEE, 2017.

[10] Sewak, Mohit, Sanjay K. Sahay, and Hemant Rathore. "Comparison of deep learning and the classical machine learning algorithm for the malware detection." 2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parall