IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 6, June 2023

DOI: 10.17148/IJARCCE.2023.12687

Behavioral Analysis and Machine Learning for Polymorphic Malware Detection and Classification / Behavior-Based Detection and Classification of Polymorphic Malware:A Machine Learning Approach

Ananth J[1], Kumaran M[2], Lin Eby Chandra J[3]

Student, Department of CSE, Jaya Engineering College, Chennai, India[1]

Professor, Department of CSE, Jaya Engineering College, Chennai, India[2]

Assistant Professor, Department of CSE, Jaya Engineering College, Chennai, India[3]

Abstract: Malware posing particular challenges. Unlike traditional malware, polymorphic variants dynamically alter their characteristics, often combining attributes from multiple malware types to evade detection by signature-based models. This article focuses on behavior-based detection and classification methods for polymorphic malware. By analyzing the behavioral patterns exhibited by malware, security professionals can develop effective detection techniques that transcend the limitations of traditional approaches. The article explores the development of behavior-based malware detection and classification methods using various machine learning algorithms. By leveraging insights obtained from static and dynamic analysis, behavioral patterns are extracted and utilized in machine learning models to predict the presence of malware and identify its malware family. Additionally, the article discusses behavior-based detection methods such as sandboxing, anomaly detection, and dynamic analysis. These techniques enable the observation and analysis of malware behavior, facilitating the identification of malicious activities and the creation of robust detection mechanisms. The findings presented in this article highlight the importance of behavior-based analysis and machine learning in combating polymorphic malware, enhancing cybersecurity measures to protect users from evolving cyber threats.

Keywords: Machine learning, detection, and classification; static analysis;

I. INTRODUCTION

The usage of the internet, computers, and smart gadgets is widespread nowadays, and many people use them on a daily basis. In the same way that there are good people and bad people everywhere we travel, the online world certainly has its share of nefarious characters that wish to use loyal users for their own gain . Malware attacks have become increasingly complicated in recent years. Malware is the most potent menace to the cyber world despite advances in detection and classification of the threat into its correct family class and ongoing evolution. Malware detection and classification are crucial because they determine which family of malware a piece of software belongs to, and on that basis, malware prevention or anti-malware solutions can be developed with a distinctive signature to identify the virus.

Malware comes in a variety of forms based on the motivation behind its creation, such as ransomware used for financial gain, spyware used for spying, etc.

We need a fundamental understanding of the sorts of malware and the tactics they employ in order to analyse malware using machine learning. Based on their behaviour, the class was separated. as described below

Virus - A virus is a programme just like any other. The primary distinction is that the programme operates on the system without the user's prior consent and replicates itself to infect other programmes on the computer.

Worm - Worms are simply an improved form of a virus. The primary distinction is that any machines connected over a network are vulnerable and could become worm-infected.

Trojan - The major goal of Trojan design is to make it appear to be legitimate software, tricking users into thinking it is safe to use.
Ransomware - Is currently the most common sort of malware. In essence, it encrypts all user data on the computer and demands a payment to restore it to working order.

DOI: 10.17148/IJARCCE.2023.12687

Adware's - Is primary objective is to display advertisements on the target computer.

Backdoor - A type of malware called a backdoor is used to create a back door for entrance into a target machine. It has little negative impact on the system.

The existing antivirus programmes primarily use signatures to detect malware. These signatures for detecting the infection are taken from malware samples that have already been gathered.If the virus has previously been identified, these signature-based solutions perform quite well, but they are unable to detect new copies of malware. Therefore, signature-based solutions aren't always enough.

New detection techniques are required to combat the threat posed by upgraded malware. Possible solutions to this issue include integrating signature-based analysis with machine learning approaches, which can produce higher accuracy than using a single signature-based strategy for detection alone.

II. OBJECTIVES

Here, the goal is to apply machine learning and create an algorithm that will successfully classify malware using machine learning in a highly accurate manner.

Enhancing the malware incident response process - The effects malware can have on a system and the precautions that need to be taken to stop it from spreading and harming the system are known if the virus's family is recognized.

Understanding how malware functions and the most recent methods used to create it is the goal of malware research. By taking into account the newly discovered features, machine learning model accuracy will be increased.

Finding new compromise indicators - Organizations can employ security solutions to better protect themselves against malware attacks by using newly discovered compromise indicators.

Track a Malware Family's Evolution - By classifying the malware according to its family, we can keep tabs on modifications and the infection's evolution over time.

III. LITERATURE SURVEY

Ying-Dar Lin, Yuan-Cheng Lai 2015 SVM model was utilized for malware family classification. Byte sequences, API and system calls, file system information, and CPU registers were features employed by the author. Information was taken out of the author's own sandbox. The approach's limitation is that accuracy is decreased as a result of evasion strategies and the need for manual intervention with samples.

Edward Raff and Charles Nicholas. 2017 The author of the paper employed a sizable data set with more than 2 lac samples, of which 1 lac were benign and the remaining were malevolent. The author only used byte sequenced files as the featured format and applied the Rule-based classifier and SVM algorithm, however because the author only used a few classes for evaluation, it was not a practical approach for classifying additional files.
Similar to the author, this work likewise utilised a sizable data collection of samples. All that could be extracted was a byte sequence.

For the data set, the author used a programme sample and some reliable apps from vxheaven. Because there are so few samples, the data set is small. Byte sequences and API system calls were employed by the author as features. The algorithms Random Forest, SVM, and Naive Bays were employed. Due to the tiny data set size for this method, many malware samples might go undetected.

Blake Anderson, Daniel Quist, Joshua Neil, Curtis Storlie, 2011, There are around 2230 samples, 615 of which are benign and the rest are malicious. For malware identification using the SVM algorithm, the author employed features such byte sequence and API/system calls. Because of the limited sample sizes employed in this study, the result accuracy is poor.

Jinrong Bai, Junfeng Wang, and Guozhong Zou ,The author used a data collection of about 20 000 samples, of which 10.5 000 were malicious and the rest were benign. Characteristic taken from PE file used by author. Although the author used the Decision Tree and Random Forest algorithms, accuracy suffers when the number of packed and updated PE header files increases.

DOI: 10.17148/IJARCCE.2023.12687

Joshua Saxe and Konstantin Berlin 2018 utilised strings and a PE file feature (character).Here, the number of harmful samples is over 4 times that of benign samples. Author classified dangerous and benign behaviour using neural networks. The above method has one drawback: the training data set's labelling may not be 100% accurate.

IV. METHODOLOGY

The focus of malware detection is on determining if a particular sample is harmful. By using malware similarity analysis, we can determine that the file is dangerous and search for attributes that will assist us accurately categorise the sample and find malware. After deciding on the analysis's eventual goal, the next stage is to extract features according to demand. Following that final stage, a machine learning method is used to attain the goal.

Goals of malware analysis - The primary goal of malware analysis is detection. Malware is discovered using a special signature that was created based on earlier samples that were appropriately identified as harmful or benign. Identifying whether a particular sample is malicious or not is always the first and foremost objective. The majority of review effort is done with the intention of finding malware.

Extraction of Features - The two basic techniques for extracting the feature from malware binary are static or dynamic analysis, either alone or in combination. The malware file is examined during static analysis without being run; all features are mostly derived from PE headers or by dissecting executable files and analysing them in assembly language. In dynamic analysis, the executable file is executed in a controlled environment, and its behaviour is observed, including any system calls that are made dynamically but are not coded, attempts to connect to any external networks, and attempts to modify registry files.

executable uses calls from the Windows API. The behaviour of an executable can be predicted, but at a higher level, depending on which API call is used. They can be taken out utilising both static and dynamic analysis techniques. The class of the executable can be determined in part using API calls. The operating system relationship can be predicted with the use of the API calls.

Strings: If a string can be parsed, it can provide information about a computer's malevolent behaviour. In a file, the strings can be encoded. The collected strings can reveal malicious code attacker intentions.

Byte sequence: The n-gram based approaches used on executable files to pick or eliminate the functions employ the byte sequence approach. In programming, N-grams are substrings of length N.N numbers range from 2-3 to 2-4, while 1 may also be used on occasion. Because the byte sequence contains executable machine code, it can be used to link some portable executable functions to resource information.

Opcodes: Opcodes are machine language instructions that will be carried out by a computer. These are nothing more than a set of instructions written in assembly language (for example, mov axe bx). The functionality of the code may be reflected in these instructions.

Decision Tree Classification and regression models are typically built using decision trees. Decision tree classifier is employed in the project to categories the virus. It created a classification model with a tree-like topology. Guided learning is what decision trees do.

Random Forest A supervised classification algorithm is Random Forest. Given that a forest can be formed by many decision trees, Random Forest is closely related to the Decision Tree algorithm. The amount of trees in the forest has a significant impact on the accuracy of the random forest model; fewer trees produce low accuracy but faster performance, whereas more trees produce better accuracy but slower execution.

Fast and lightweight, the Light GBM algorithm learns based on the structure of trees. Regression and classification are the two main applications of gradient boosting. It creates a new iterative model from a combination of various algorithms.

DOI: 10.17148/IJARCCE.2023.12687

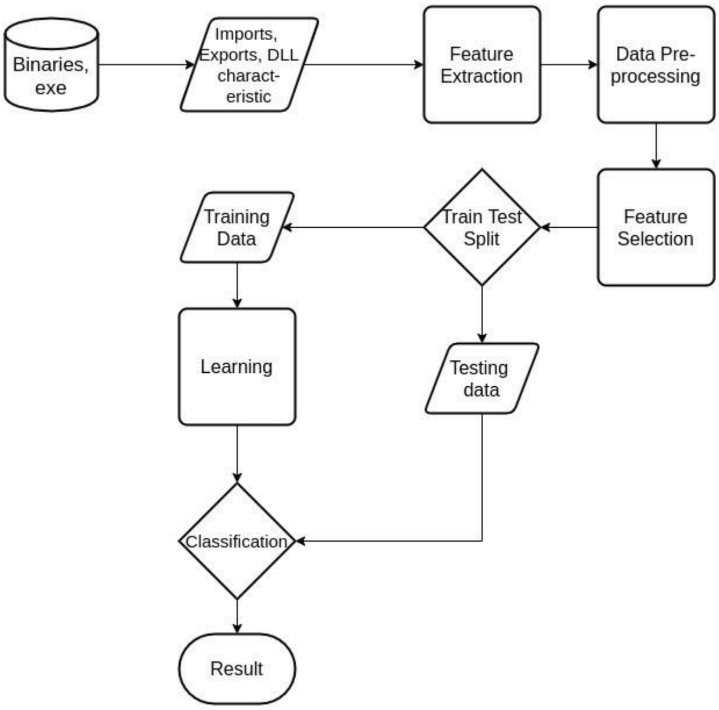

Fig 1: Architecture

Figure 1 depicts the proposed architecture for malware detection and classification. The total process flow is separated into the following steps based on the architecture

1) Create a data set first. With the aid of a PE file and a Python library, extract the static data from the programme or software and produce Excel spreadsheet preserving data for each programme or piece of software

2) Data preparation Preparing the data is a crucial step before choosing the features for the model. Remove any null values from the data set during data pre-processing. Remove the columns that contain the categorical information.

3) Selection of Features Choose the features that are crucial for the output or for accurately predicting or classifying the input. The data set should be divided into two halves for training and testing purposes after the suitable characteristics have been chosen.

4) Training Upon successful completion of the previous step, train the preferred algorithm for classification using the training data set.

5) Classification This final phase involves predicting the actual classification using the training from the previous step. The method chosen for classification makes a prediction about whether a sample of data is malicious or benign.

V. CONCLUSION

The static features that were retrieved from both good and bad executable files were used by the machine learning method. This method makes it very quick to determine whether a given file is harmful or not. To lessen the burden on dynamic analysis of executables in heavy load conditions, it will be effective to apply this approach before signature-based solutions. For each model employed in classification, different results were obtained. While Random Forest has superior accuracy, Decision Tree has a lesser accuracy of 99.14%. The algorithm developed by Light Gradient Boosting Machine has the highest accuracy of all of them, at roughly 99.50%, only a little bit more than random forest technique. A gradient of light The boosting machine algorithm is effective in both accuracy and model training time. In comparison to Light GBM, Random Forest is significantly slower. Light GBM has the lowest False Negative (predicting malicious as legitimate) rate out of all of them, which is a positive factor to take into account when choosing the principal algorithm for classification. Although it is recommended that false negatives in these investigations be nil or almost zero. If the rate of false negatives is larger, the model is useless in a production setting.

DOI: 10.17148/IJARCCE.2023.12687

REFERENCES

[1]. (Senior Member, IEEE),Intelligent Vision-Based Malware Detection and Classification Using Deep Random Forest Paradigm, IEEE Access ,November 6, 2020.

[2]. M. Nisa, J. H. Shah, S. Kanwal, M. Raza, M. A. Khan, R. Damaše vičius, and T. Blažauskas, ‘‘Hybrid malware classification method using segmentation-based fractal texture analysis and deep convolution neural network features,’’ Appl. Sci., vol. 10, no. 14, p. 4966, Jul. 2020, doi: 10.3390/app10144966. S. Zhang, C. Zhu, J. K. O. Sin, and P. K. T. Mok, “A novel ultrathin elevated channel low-temperature poly-Si TFT,” IEEE Electron Device Lett., vol. 20, no. 2, pp. 569–571,1999.

[3]. D. Zou, Y. Wu, S. Yang, A. Chauhan, W. Yang, J. Zhong, S. Dou, and H. Jin, ‘‘IntDroid: Android malware detection based on API intimacy analysis,’’ ACM Trans. Softw. Eng. Methodology, vol. 30, no. 3, pp. 1–32, May 2021, doi: 10.1145/3442588.

[4]. O. Aslan and A. A. Yilmaz, ‘‘A new malware classification frame work based on deep learning algorithms,’’ IEEE Access, vol. 9, pp. 87936–87951, 2021.

[5]. Antono poster 1 , Alberto Mozo 2 , Stanislav Vakaruk 2 , Daniele Canavese 3 , Diegor. López 1 , Leonardo Regano3 , Sandra Gómez Canaval2 , and Antonio lioy3 , (Member, IEEE), Detection of Encrypted Cryptomining Malware Connections With Machine and Deep Learning, IEEE Access , August 26, 2020.

[6]. Hanxum Zhou 1 , Xilin Yang 1 , Hong Pan 2 , and WEI GUO 3, An Android Malware Detection Approach Based on SIMGRU, IEEE Access , July 29 2020.

[7]. Tzu-Ling Wan 1, Tao Ban 3 (Member, IEEE), Shin-Ming Cheng 1,2 (Member, IEEE), Yen-Ting Lee1, BO Sun4, Ryoichi ISAWA3, Takeshi Takahashi3 (Member, IEEE), Efficient Detection and Classification of Internet-of-Things Malware Based on Byte Sequences from Executable Files, IEEE Access ,26 October 2020.

[8]. 4 Durmus Özkan Şahin , Sedat Akleylek , LinRegDroid: Detection of Android Malware Using Multiple Linear Regression Models-Based Classifiers , IEEE Access , January 27, 2022.

[9]. Joshua Saxe and Konstantin Berlin. Deep neural network based malware detection using two dimensional binary program features. In 2015 10th International Conference on Malicious and Unwanted Software (MALWARE), pages 11–20. IEEE, 2015

[10]. Daniele Ucci, Leonardo Aniello, and Roberto Baldoni. Survey of machine learning techniques for malware analysis. Computers & Security, 81:123–147, 2019.