Engineering and Technology Research 5(4): 063-065, July 2022
DOI: 10.15413/etr.2022.0107

ISSN: 2682 5716

Research Paper

Prediction of malware analysis using machine learning and deep learning techniques

Accepted 12th May, 2022

ABSTRACT

The term malware refers to software that harms or exploits devices and computers. Corporations and government agencies struggle with malware, including viruses, worms, Trojan horses, ransomware, and spyware. The traditional method of detecting malware relies on anti-virus signatures, heuristics, and sandbox testing, which require manual analysis by security analysts and researchers (Owaida, 2021). Organizations have a hard time keeping up with malwarethreats daily as new attacks and variants emerge. Machine learning (ML) and artificial intelligence (AI) can detect unknown malware by automatically learning the malware patterns based on large volumes of historical data. With its unique capabilities, artificial intelligence/machine learning has become an integral part of the latest malware detection solutions, complementing heuristics and signature-based methods. This study attempts to construct Malware detection algorithms based on multiple machine learning and deep learning techniques using Python in Google Colab that cansignificantly identify the difference between malicious and non-malicious files. This proposed method tries to take a sample dataset collected from the Kaggle.com website as input, and then the implementation of the pre-processing step is done, followed by applying the concepts of ML and deep learning to predict the malware and reduce the false positives. As a result, RBM and Hybrid models turned out to give the best results with reduced test loss in the case of RBM.

Maheswari Kota

Gandhi Nagar, Rushikonda, Visakhapatnam, Andhra Pradesh 530045, India.

E-mail: mkota@gitam.in.

Key words: Malware, Machine learning, deep learning, artificial intelligence.

INTRODUCTION

All database servers are vulnerable to a slew of new security threats in today's world. At present, Malware is being spread throughout the Internet and has become an indispensable aspect of our lives because it practically connects us to the rest of the world. Malware is one of the severe threats and one of the most common cyber-attack strategies directed at businesses. This fact has given rise to a new cyber-crime business model known as ransomware, which allows an attacker to encrypt an organization important data and files stored on servers, storage area networks (SANs), and endpoint devices. Only after the ransom is paid the victim be given the decryption key.

Currently, ransomware is a significant cyber threat aimed at individuals and organizations alike, specifically at organizational VMs in the computational cloud. Ransomware usually prevents or limits access to resources (such as data and services) in the infected host (Kim and Kim, 2018; Kirda, 2017). According to recent information from Kaspersky Lab, cyber-attacks launched by ransomware aimed at businesses tripled in 2016, and as of October 2016, it was reported that ransomware hit a new victim every 40 s.

One Malware may have multiple characteristics. By leveraging Malware, attackers could attack kernel, control

remote computers and continue to attack other computers. With the development of cloud computing, virtual machines (VMs) have become one of the significant targets of Malware.

A standard attack vector exploits vulnerabilities in web services and then lets virtual machines run malicious code to attack other VMs. Therefore, malware detection is essential to cloud security. To detect malware, static and dynamic analysis approaches are proposed to analyze malware signatures. Static analysis tools analyze binary images to detect Malware. Firstly, Malware is unpacked and decrypted. After that, feature extraction analyzes Malware, such as string signature, control flow graph, and some specific APIs' statistics. Dynamic analysis approaches extract features during malware execution. They employ many dynamic tracing technologies and features to analyze Malware, such as function tracing, API tracing, file tracing, and network traffic. Machine learning technology is usually employed to build models based on the extracted features to detect Malware. Machine learning algorithms include Support Vector Machine (SVM), N-gram, decision tree, and neural networks. Advanced methods such as machine learning algorithms have demonstrated capability and effectiveness in detecting Malware (both known and unknown) on a variety of platforms (Finsinger and Tinner, 2007; Christodorescu and Jha, 2004; White et al., 1999; Kharraz et al., 2015; Sittig and Singh, 2016; Nissim et al., 2014; 2016a,b; 2016a,b; Moskovitch et al., 2009; Cohen et al., 2016, 2018). However, to the best of our knowledge, the MinHash (Broder, 1997) method has not been used to secure efficient and trusted detection of unknown Malware residing on a VM.

BACKGROUND

Malware evolution

It is challenging to secure computers and networks from attack due to the diversity, sophistication, and availability of malicious software. Malware is continually evolving, so security analysts and researchers must continue improving cyber defenses to keep pace. Polymorphic malware mutates the original code while maintaining actual functionality as it uses polymorphic-engine technology to evade detection and hide its true purpose. Metamorphic malware uses some metamorphic techniques to avoid detection and conceal its meaning. There are two main methods of hiding code: packing and encryption. Packing hides the actual code through layers of compression, and then at runtime, unpacking routines restore the existing code.

Metamorphic malware rewrites its code at any time to an equivalent style when it is propagated. A crypter encrypts and decrypts malware or part of it to disguise it from researchers. A crypter generally has a stub that encrypts and decrypts malicious code. Malware authors may use several transformation techniques: register renaming, code permutations, code expansion, shrinking, and garbage code insertion. By combining the above methods, malware volumes overgrew, making malware cases more time-consuming, costly, and challenging for forensic investigations.

The traditional antivirus solutions that rely on signatures or heuristic/behavioral methods have their problems. A signature differs from an executable like a fingerprint because it contains features that make it unique. Although signatures can detect known malware variants, they are not capable of detecting unknown malware. To overcome these challenges, security analysts developed behavior-detection identifies to determine whether or not a file is malware based on its characteristics and behavior, though the scanning and analysis process takes time. Machine learning was introduced as a complement to traditional antivirus engines to overcome the limitations of conventional antivirus engines and keep pace with new attacks and variants (Gibert et al., 2020). Since machine learning is well suited to processing large amounts of data, researchers adopted this technology to remain competitive and keep up with new threats.

Machine learning approaches

As a result of machine learning solutions being developed and deployed over the past decade, malware detection and classification have become increasingly sophisticated. It was only through the convergence of three recent developments that machine learning approaches have managed to succeed and consolidate:

I. The first development is the proliferation of labeled malware feeds that now allow the security community and the academic community to gain access to labeled malware for the first time.
II. Secondly, computational power has increased rapidly while, at the same time, it has become more affordable and within reach of the majority of researchers. As a result, researchers could speed up their iterative training processes and fit more immense and more complex models to increasingly large data samples.
III. Third, machine learning has experienced significant growth over the past decades, establishing new standards for accuracy and scalability and achieving breakthrough success on a range of tasks, including computer vision, speech recognition, and natural language processing.

During a machine learning workflow, available data are gathered, cleaned and prepared, models are built, models are validated, and deployments are made. A traditional machine learning approach uses pre-processing rather than

Table 1: The attained accuracies.

Model used	Accuracy obtained
Bagging	0.97
Gradient Boost & AdaBoost	0.98
MLP classifier	0.83
Hybrid Model	0.99
Restricted Boltzmann Machine	0.99

malware to provide an abstract view of the software by extracting a set of features from the executable.

A model is trained using these features to solve the task at hand. Because of the wide variety of functions provided by malware, it is crucial to identify malicious software and determine how it differs from different types of Malwares.

Machine learning solutions differ mainly in the outcome returned by the implemented system to detect or classify malware. One way to determine the maliciousness of an executable is to calculate the value y = f(x) from 0 to 1, which is a single numeric value. Alternatively, a classification system produces the probability of a given executable belonging to a given outputclass or family, y E R*, where N represents the number of different families.

Datasets

There are mainly system files (from different versions of operating systems) and executable files from popular applications in the training database.

A Virus Total collection files have been included in the dataset. The two types of files in the sample are malware files and clean files.

METHODS

Our study uses machine learning techniques to determine the best outcome topredict the accuracy with the decreased rate of false negatives of predicting an infected file. The algorithms implemented in our methodology:

- Ensemble methods (Bootstrap Aggregating, Boosting),
- Neural networks: Feed Forward Neural Network, Recurrent neural network, and a network using Multi-layer Perceptron classifier,
- Restricted Boltzmann Machine,
- Hybrid model.

In the first step of implementation, we have combined CSV files in which filenames are labelled as 0 or 1 (for benign or malicious, respectively). Afterward, the dataset is checked against the null values, for which we used simple interpolation methods and are also used to learn from the incomplete data. The interpolation methods are chosen so that the Nan, that is, the NULL values, do not affect the machine learning algorithms' analysis or accuracy. The methods are Mean (which is the average of all values), Median (it is the middle value of a set of sized values), Mode (a number that occurs more than twice or thrice in a set of given data), and in our case, we took the median.

We are now applying the algorithms mentioned above. The dataset is significant and must be reduced without any data loss. Hence Principal Component Analysis (PCA) is used, the columns are reduced to 55, and a random forest classifier is applied in the ensemble method. Scaling numerical input variables to a standard range improves the performance of many machine learning algorithms. Before modelling, data are normalized and standardized using two of the most popular techniques (Brownlee, 2020). The **normalization** scales each input variable from 0-1, the range for floating-point values where we have the most accuracy. The **standardization** process removes the mean from each input variable (called centring) and divides the difference by thestandard deviation to shift the distribution to a mean of zero and a standard deviation of one. The ensemble method is implemented with the help of the decision tree classifier and the random forest classifier to obtain the results as (0.970, 0.964) and (0.984, 0.983) for bagging and boosting (AdaBoost) classifiers, respectively. Even after implementing the Gradient Boost, the accuracy remained the same, that is, 0.98 (Table 1).

Furthermore, for the neural networks, the accuracy ranged from 0.825 to 0.969, whereasthe Hybrid model and the Restricted Boltzmann machine gave 0.99.

CONCLUSION

In the present study, a proof-of-concept malware detection method was created. The performance metrics for the different classifiers used are also presented, with the Hybrid modelhaving an accuracy of 99%. Analyzes of the results of the tests showed that this proof-of- concept is quite effective and efficient in detecting malware.

REFERENCES

Brownlee J (2020). In Data Preparation.

Gibert D, Mateu C, Planes J (2020). The rise of machine learning for detection and classification of malware: Research developments, trends and challenges.

Owaida, 2021.