Shairoze *et al.LGU (IJECI)* 2020

LGU (IJECI)
ISSN: 2522-3429 (Print)
ISSN: 2616-6003 (Online)

LGU International Journal for
Electronic Crime Investigation

Research Article

Vol. 5 issue 3 Year 2021

Machine Learning in Malware Detection

Muhammad Shairoze Malik*
13beemmalik@seecs.edu.pk
National University of Science
and Technology Islamabad

Abstract:

Malware has become one of the biggest cyber threats today with the rapid growth of the Internet. Malware can be referred to as any program that performs malicious acts, including data theft, espionage, etc. In a world of growing technology, protection should also increase at the same time. Machine learning has played a significant role in operating systems over the years. Cybersecurity is capable of using machine learning to boost organizations' detection of malware, triage, breach recognition and security alert. Machine learning will significantly change the cyber security climate. New techniques such as machine learning must be used to solve the rising malware problem. This paper aims to research how cybersecurity can be used for machine learning and how it can be used to detect malware. We will look at the PE (portable executable) headers of samples of malware and non-malware samples and create a classifier for malware that can detect whether or not malware is present.

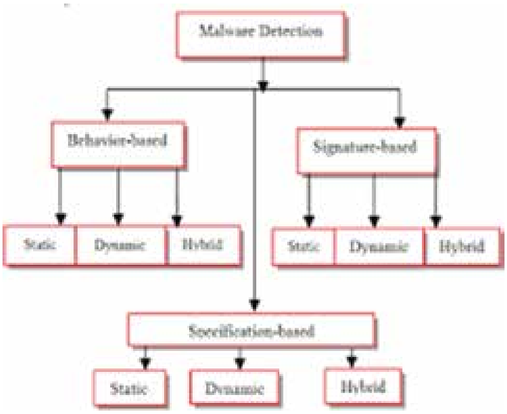
Keywords: Cybersecurity, detection, malware, machine learning, PE headers, classifier, preparation, boost

1. Introduction

Idealistic hackers attacked computers in the early days because they were eager to prove themselves. Cracking machines, however is an industry in today's world. Despite recent improvements in software and computer hardware security, both in frequency and sophistication, attacks on computer systems have increased. Regrettably, there are major drawbacks to current methods for detecting and analyzing unknown code samples. The Internet is a critical part of our everyday lives today. On the internet, there are many services and they are rising daily as well. Numerous reports indicate that malware's effect is worsening at an alarming pace. Although malware diversity is growing, anti-virus scanners are unable to fulfil security needs, resulting in attacks on millions of hosts. Around 65,63,145 different hosts were targeted, according to Kaspersky Labs, and in 2015, 40,00,000 unique malware artefacts were found. Juniper Research (2016), in particular, projected that by 2019 the cost of data breaches will rise to $2.1 trillion globally [1]. Current studies show that script-kiddies are generating more and more attacks or are automated. To date, attacks on commercial and government organizations, such as ransomware and malware, continue to pose a significant threat and challenge. Such attacks can come in various ways and sizes. An enormous challenge is the ability of the global security community to develop and provide expertise in cybersecurity. There is widespread awareness of the global scarcity of cybersecurity and talent. Cybercrimes, such as financial fraud, child exploitation online and payment fraud, are so common that they demand international 24-hour response and collaboration between multi-national law enforcement agencies [2]. For single users and organizations, malware defense of computer systems is therefore one of the most critical cybersecurity activities, as even a single attack may result in compromised data and sufficient losses. This research explores how machine learning can be used in the field of cybersecurity, along with how it can be used to detect malware. In order to detect malware, we will examine the PE headers of malware and non-malware samples or files by creating and training a classifier that will determine whether the file has been attacked by malware or not after training.

2. Evolution of Malware

In order to protect networks and computer systems from attacks, the diversity, sophistication and availability of malicious software present enormous challenges. Malware is continually changing and challenges security researchers and scientists to strengthen their cyber defenses to keep pace. Owing to the use of polymorphic and metamorphic methods used to avoid detection and conceal its true intent, the prevalence of malware has increased. To mutate the code while keeping the original functionality intact, polymorphic malware uses a polymorphic engine. The two most common ways to conceal code are packaging and encryption [3]. Through one or more layers of compression, packers cover a program's real code. Then the unpacking routines restore the original code and execute it in memory at runtime. To make it harder for researchers to analyze the software, crypters encrypt and manipulate malware or part of its code. A crypter includes a stub that is used for malicious code encryption and decryption. Whenever it's propagated, metamorphic malware rewrites the code to an equivalent. Multiple transformation techniques, including but not limited to, register renaming, code permutation, code expansion, code shrinking and insertion of garbage code, can be used by malware authors. The combination of the above techniques resulted in increasingly increasing quantities of malware, making time-consuming, expensive and more complicated forensic investigations of malware cases. There are some issues with conventional antivirus solutions that rely on signature-based and heuristic/behavioral methods. A signature is a unique feature or collection of features that like a fingerprint, uniquely differentiates an executable. Signature-based approaches are unable to identify unknown types of malware, however. Security researchers suggested behavior-based detection to overcome these problems, which analyses the features and behavior of the file to decide whether it is indeed malware, although it may take some time to search and evaluate. Researchers have begun implementing machine learning to supplement their solutions in order to solve the previous drawbacks of conventional antivirus engines and keep pace with new attacks and variants, as machine learning is well suited for processing large quantities of data. [4]



3. Malware Detection

In such a way, hackers present malware aimed at persuading people to install it. As it seems legal, users also do not know what the program is. Usually, we install it thinking that it is secure, but on the contrary, it's a major threat. That's how the malware gets into your system. When on the screen, it disperses and hides in numerous files, making it very difficult to identify. In order to access and record personal or useful information, it may connect directly to the operating system and start encrypting it [5]. Detection of malware is defined as the search process for malware files and directories. There are several tools and methods available to detect malware that make it efficient and reliable. Some of the general strategies for malware detection are:

i. Signature-based

ii. Heuristic Analysis

iii. Anti-malware Software

iv. Sandbox

Several classifiers have been implemented, such as linear classifiers (logistic regression, naive Bayes classifier), support for vector machinery, neural networks, random forests, etc.

Through both static and dynamic analysis, malware can be identified by:

✓ Without Executing the code

✓ Behavioral Analysis

4. Need for Machine Learning in Malware Detection

Machine learning has created a drastic change in many industries, including cybersecurity, over the last decade. Among cybersecurity experts, there is a general belief that AI-powered anti-malware tools can help detect modern malware attacks and boost scanning engines. Proof of this belief is the number of studies on malware detection strategies that exploit machine learning reported in the last few years. The number of research papers released in 2018 is 7720, a 95 percent rise over 2015 and a 476 percent increase over 2010, according to Google Scholar,1. This rise in the number of studies is the product of several factors, including but not limited to the increase in publicly labelled malware feeds, the increase in computing capacity at the same time as its price decrease, and the evolution of the field of machine learning, which has achieved ground-breaking success in a wide range of tasks such as computer vision and speech recognition [6]. Depending on the type of analysis, conventional machine learning methods can be categorized into two main categories, static and dynamic approaches. The primary difference between them is that static methods extract features from the static malware analysis, while dynamic methods extract features from the dynamic analysis. A third category may be considered, known as hybrid approaches. Hybrid methods incorporate elements of both static and dynamic analysis. In addition, learning features from raw inputs in diverse fields have outshone neural networks. The performance of neural networks in the malware domain is mirrored by recent developments in machine learning for cybersecurity. [6]

5. Detailed Design

Our paper workflow is divided into 3 sections.

- Describing the details: The dataset is imported and the different columns are discussed in the dataset.
- Data cleaning: The required steps are taken after examining the dataset so that the dataset can be cleaned and all the null values and columns of not much significance are removed so that they will not be of any concern in the training part.
- Data Training and Testing: When the information is transparent and ready for training, we spilled the information as a training dataset and testing dataset in an 80:20 ratios so that the data was spilled in an 80:20 ratios.

In this paper, as we try to achieve the highest accuracy, we use two algorithms to see which will give us better precision.

- Gradient Boost Classifier
- Random Forest Classifier

6. Algorithms

Gradient Boosting- Gradient boosting is a technique of machine learning which uses regression and classification problems that helps us generate a prediction model in the form of an ensemble of the weaker prediction models, usually decision trees. As other boosting techniques do it constructs the model in a phase-wise fashion and generalizes them by allowing an arbitrary differentiable loss function to be optimized. [7] For predictive model growth, gradient boosting is one of the most effective techniques. Gradient Boosting is teaching several models steadily, additively and sequentially. With gradients of loss function, gradient boosting takes place. What we strive to develop and maximize depends on a simple understanding of the loss function.

- In Gradient Boosting, three elements area feature for loss to be optimized or enhanced.
- A poor man who has learned to make predictions
- A supplementary model for incorporating disadvantaged students to minimize losses. [8][9]

It's important that we understand how the algorithm of Gradient Boost is implemented under the hood.

1. Calculate average of target label- We begin with a leaf that is the average value of the variable we want to forecast when solving regression problems. This leaf will be used in the procedural steps as a baseline to reach the correct solution.

2. Calculate the residuals- Calculating the residual with the proceeding formula.

   Residual = actual value – predicted value

3. Construct a decision tree- Next with the intention of predicting the residuals, we build a tree. In other words, a prediction of the residual value (not the desired label) will be found in every leaf. Any residuals will end up within the same leaf in the event that there are more residuals than leaves. We compute their average and position that inside the leaf when this happens.

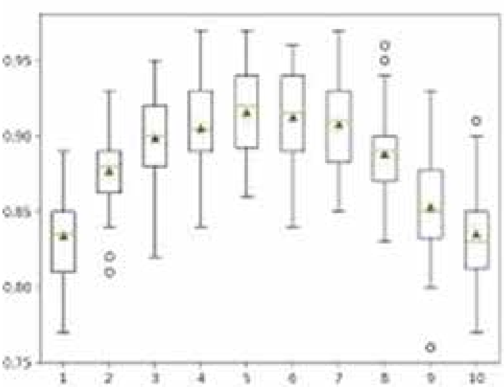
4. Using the trees within the ensemble predict the target label. - Each sample passes through the newly developed tree's decision nodes before it reaches a given lead.

5. Compute the new residuals- The residuals will then be used as explained in step 3 for the leaves of the coming next decision tree.

6. Repeat steps 3 to 5 until the number of iterations matches the number (i.e. the number of estimators) defined by the hyper parameter.

7. To make a final prediction as to the value of the target variable, use all the trees in the ensemble once eligible. [10]

In the first step, the final forecast will be equal to the mean we determined, plus all the residuals predicted by the trees that make up the forest multiplied by the learning rate.



Random forests - Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that function by constructing a multitude of decision trees at training time and generating the class that is the class mode (classification) or the individual trees' mean/average prediction (regression) of the individual trees. Random forests are often used as "Blackbox" models in companies, as they produce rational predictions over a large range of data while requiring little configuration in packages such as sci-kit-learn [11]. However, data characteristics can affect their performance. In the steps and diagrams below the working procedure can be explained:

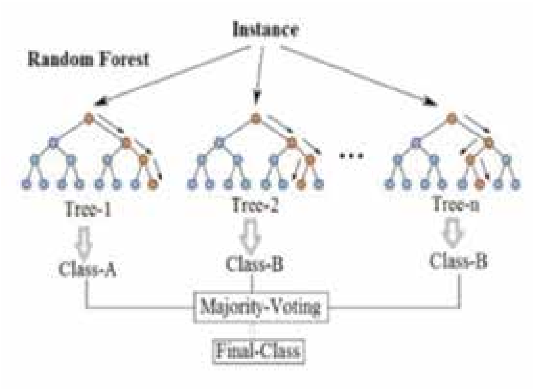
1. From the training set select random K data points.

2. Use the selected data points to build decision trees associated with it (Subsets).

3. Choose a number N which represents the decision trees that you want to build.

4. Repeat Step 1 & 2.

5. Find the predictions for new data points for each decision tree and assign the new data points to the group that receives the majority votes. [12]

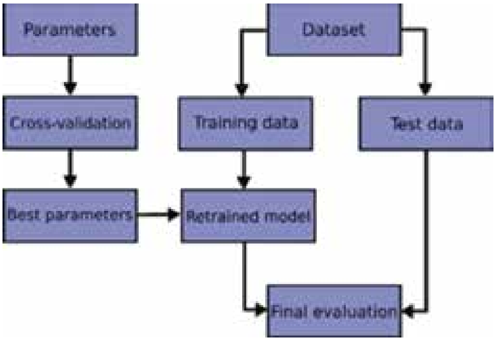


7. Mplementation

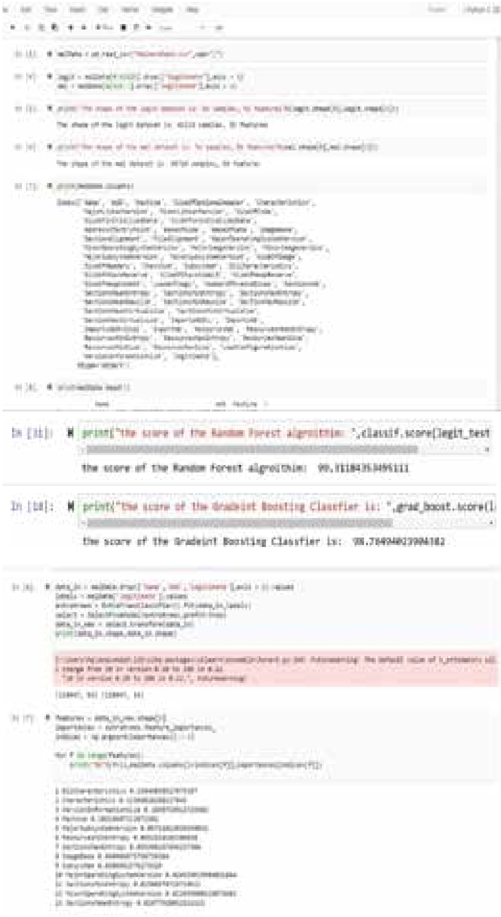
We used a dataset that was made available from the Chiheb Chebbi - Mastering Machine Learning for Penetration Testing book for this work. There are approximately 138000 entries of legit and malware PE headers and 56 columns as features in the dataset. In an 80 percent preparation and 20 percent evaluation, the knowledge was divided.

We initially import and read the dataset, once that is done, we clean the dataset by dropping unnecessary features and null values. After that we split the dataset for training and testing. We import the necessary packages for making a decision tree, gradient boosted classifier and random forest classifier. Once done we fit that data respectively and predict the results.

Using a combination of these algorithms, after training and testing the algorithms, we were able to get a highly accurate outcome.



8. Results



```
[illegible]

In [31]:  print("the score of the Random Forest algroithm: ",classif.score(legit_test

          the score of the Random Forest algroithm:  99.31184353495111

In [18]:  print("the score of the Gradeint Boosting Classfier is: ",grad_boost.score(l

          the score of the Gradeint Boosting Classfier is:  98.76494023904382

[illegible]
```

After training and testing both the algorithms, we can see that both of them give us a high accuracy output.

9. Conclusion

The algorithm used for training the data was Gradient Boosted classifier and random forest classifier which gives us an accuracy of 98.764% and 99.311% respectively. After viewing the confusion matrix of the random forest classifier, we could conclude that the number of false positives were at 0.5505 and false negatives were at 1.0053. And after viewing the confusion matrix of the Gradient boosted algorithm we can say that the number of false positives were at 0.768 and false negatives were at 2.3099.

Our main objective was to come up with a system for machine learning that typically detects as many samples of malware as possible, with the tough restriction of having a zero false positive rate. We have been really close to our target, but we still have a false positive rate that is non-zero. A variety of deterministic exemption mechanisms must be added in order for this system to become part of a highly competitive commercial product. In our view, machine learning detection of malware will not replace the existing methods of detection used by anti-virus vendors, but will come as an extension to them. Certain speed and memory limitations are placed on any commercial anti-virus product, so the most accurate algorithms should be used.

10. References

[1] Ahmadi etal., 2016. M. Ahmadi, D. Ulyanov, S. Semenov, M. Trofimov, G. Giacinto - Novel feature extraction, selection and fusion for effective malware family classification

[2] AL-Hawawreh etal., 2018 M. AL-Hawawreh, N. Moustafa, E. Sitnikova - Identification of malicious activities in industrial internet of things based on deep learning models

[3] Athiwaratkun etal., 2017 B. Athiwaratkun, J.W. Stokes - Malware classification with lstm and gru language models and a character- level cnn

[4] D. Bekerman, B. Shapira, L. Rokach, A. Bar - Unknown malware detection using network traffic classification 09 2015

[5] B. Biggio, F. Roli - Wild patterns: ten years after the rise of adversarial machine learning

[6] I. Santos, Y. K. Penya, J. Devesa, and P. G. Garcia, “N-grams- based file signatures for malware detection,” 2009.

[7] K. Rieck, T. Holz, C. Willems, P. D¨ussel, and P. Laskov, “Learning and classification of malware behavior,” in DIMVA ’08: Proceedings of the 5th international conference on Detection of Intrusions and Malware, and Vulnerability Assessment. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 108–125.

[8] E. Konstantinou, “Metamorphic virus: Analysis and detection,” 2008, Technical Report RHUL-MA-2008-2, Search Security Award M.Sc. thesis, 93 pages.

[9] Gibert etal., 2019 D. Gibert, C. Mateu, J. Planes - A hierarchical convolutional neural network for malware classification. The International Joint Conference on Neural Networks 2019, IEEE (2019), pp. 1-8

[10] X. Guo, Y. Yin, C. Dong, G. Yang, G. Zhou - On the class imbalance problem 2008 Fourth International Conference on Natural Computation, vol. 4 (Oct 2008), pp. 192-201

[11] Hall, 1999 M.A. Hall - Correlation-based Feature Selection for Machine Learning Ph.D. thesis The University of Waikato (1999)

[12] W. Han, J. Xue, Y. Wang, L. Huang, Z. Kong, L. MaoMaldae: - Detecting and explaining malware based on correlation and fusion of static and dynamic characteristics Comput. Secur., 83 (2019), pp. 208-233

[13] W. Han, J. Xue, Y. Wang, Z. Liu, Z. KongMalinsight: a systematic profiling-based malware detection framework J. Netw. Comput. Appl., 125 (2019), pp. 236-250

[14] X. Hu, K.G. Shin, S. Bhatkar, K. Griffin Mutantx-s: scalable malware clustering based on static features Presented as Part of the 2013 USENIX Annual Technical Conference (USENIX ATC 13), USENIX, San Jose, CA (2013), pp. 187-198

[15] Huang and Stokes, 2016 W. Huang, J.W. StokesMtnet: a multi- task neural network for dynamic malware classification Caballero J., Zurutuza U., Rodrguez R.J. (Eds.), Detection of Intrusions and Malware, and Vulnerability Assessment, Springer International Publishing, Cham (2016), pp. 399-418

[16] X. Zhang, J. Zhao, Y. LeCun Character-level convolutional networks for text classification Proceedings of the 28th International Conference on Neural Information Processing Systems, ume 1, MIT Press, Cambridge, MA, USA (2015), pp.

[17] D. Uppal, R. Sinha, V. Mehra, V. Jain - Malware detection and classification based on extraction of api sequences 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (Sep. 2014)

[18] A. Souri, R. Hosseini - A state-of-the-art survey of malware detection approaches using data mining techniques Human-centric Computing and Information Sciences, 8 (1) (Jan 2018),

[19] I E. Raff, J. Barker, J. Sylvester, R. Brandon, B. Catanzaro, C.K. Nicholas - Malware detection by eating a whole EXE - The Workshops of the the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2- 7, 2018 (2018)

[20] A. Moser, C. Kruegel, E. Kirda - Limits of static analysis for malware detection Twenty-Third Annual Computer Security Applications Conference (ACSAC 2007) (Dec 2007), pp. 421-43