A Supervised Machine Learning Algorithm for Detecting Malware

Olaniyi Abiodun Ayeni
Department of Cyber Security, School of Computing
Federal University of Technology, Nigeria

Abstract

The proliferation of malware is a threat to our computing system and its security. That is why the need for malware detection using machine learning arises. This work was motivated by the limitation of [1], [2] in 'Malware Detection Module using Machine Learning Algorithms. The objective of this research is to develop a security system for the detection of malware using supervised machine learning algorithms and also to carried out performance evaluation. Feature selection (Filter method) was used to reduce 100,000 columns and 35 rows of features to 20 features, then three classifier algorithms were employed which are K-Nearest Neighbor, Decision Tree and Random Forest. The classifiers are trained and tested using the dataset(malware.csv) gotten from Malware Detection Kaggle. The results of the algorithms (K-Nearest Neighbor, Decision Tree and Random Forest) are respectively 96.53%,97.79% and 99.90%. The results were also compared with other researchers[3] work that used the same three classifiers, the results of Maqsood 2020 for Random Forest, Decision tree and K nearest neighbor are respectively 96.39%, 100%(overfit) and 99.4%, while the results of Sarang et al 2013 for Random Forest, Decision tree and K nearest neighbor are respectively 99.57%, 99.23%, and 99.06%. It indicates that Random Forest is most effective out of the three classifiers algorithm for malware detection using machine learning, moreover, the study performed can be useful as a base for further research in the field of malware analysis with machine learning methods.

Keywords: Malware, Supervised Learning, Decision Tree, K-Nearest Neighbour, Random Forest, Feature Method, Computer Security

1. Introduction

The malware was first created in 1949 by John von Neumann. Ever since then, more were created. Antivirus company is continually searching for the most effective ways in identifying malware and one of the most famous methods used is the signature-based detection. Furthermore, the skill level that is required for malware development is on the decrease because of the high numbers of attacking tools on the internet nowadays. High availability of anti-detection techniques and the ability to buy malware in the black-market results in an opportunity to be an attacker for anyone, not depending on the skill level. Current studies show that more attacks are being issued by script-kiddies or automated [4]. Therefore, malware protection of computer is an essential cybersecurity tasks for single users and businesses, since an attack can lead to compromised data and sufficient losses. Also, massive losses and frequent attacks influences the need for accurate and timely detection methods. However, current static and dynamic methods do not provide efficient detection, especially when dealing with zero-day attacks. Hence, machine learning-based techniques can be used. Therefore, this paper discusses the main points and concerns of machine learning-based malware detection, as well as looks for the best feature representation and classification methods.

2. Research Motivation

The existing literature on the topic of malware detection convinces that there is a need for efficient malware detection system, especially since the use of internet are becoming increasingly important nowadays.

The existing frameworks Malware Detection Module using Machine Learning Algorithms to Assist in Centralized Security in Enterprise Networks' [2] focuses on just the detection and classification neglecting home users because it's processor heavy, also in Detection of malicious code by applying machine learning classifiers on static features Shabtai et al. [5] models were not trained properly which resulted to running inefficient algorithms. This provided the motivation to create a malware detection system using machine learning that is well trained and has a high accuracy and low positive rate using machine learning that can protect one's system by flagging incoming malicious files and preventing them from affecting one's computer.

2.1. Problem Statement

With the development of technology, the number of malwares is increasing daily. Malware is now designed with mutation characteristic, which causes an enormous growth in the number of variations. Also, with the help of automated malware generated

tools, novice malware authors can now quickly generate a new variation. With these growths, traditional signature-based malware detection is proven to be ineffective against the vast variation of malware. However, machine learning methods for malware detection has proven effective against new malware.

Objective: The objectives of this research work are to:

- Design a security framework for malware detection using Supervised Machine learning algorithm.
- Implement the design in (a).
- Evaluate the performance of the system.

2.2. Malware Types

i. *Backdoor*: It is a malware type that negates standard authentication procedures to access a system. As a result, remote access is granted to resources within an application, such as databases and file servers, giving perpetrators the ability to issue system commands remotely and update malware.

ii. *Rootkit*: Its functionality enables the attacker to access the data with higher permissions than is allowed. For example, it can be used to give unauthorized user administrative access. Rootkits always hide their existence and quite often are unnoticeable on the system, making the detection and, therefore, removal incredibly hard. [6].

iii. *Keylogger*: The idea behind this malware class is to log all the keys pressed by the user and store all data, including passwords, bank card numbers, and other sensitive information [7].

iv. *Ransomware*: This type of malware aims to encrypt all the data on the machine and ask a victim to transfer some money to get the decryption key. Usually, a machine infected by ransomware is "frozen" as the user cannot open any file, and the desktop picture is used to provide information on the attacker's demands [8].

3. Related Works

In state-of-the-art survey of malware detection approaches using data mining techniques [1] present a survey of malware detection approaches divided into two categories:

- Signature-based methods
- Behavior-based methods.

However, the survey does not provide either a review of the most recent deep learning approaches or a taxonomy of the types of features used in data mining techniques for malware detection and classification.

The research is motivated by a serious threat today called malicious executables. It is designed to damage computer system and some of them spread over network without the knowledge of the owner using the system.

Objective(s): To present a survey of malware detection approaches.

3.1. Methodology

Provision of summary of the current challenges related to malware detection approaches in data mining, presenting a systematic and categorized overview of the current approaches to machine learning mechanisms in the data mining topics, exploring a structure of the important methods that are significant in malware detection approach, discussing the important factors of classification malware approaches in the data mining to improve their problems in the futures.

3.2. Contribution to Knowledge

It enlightens more on how to approach malware detection using machine learning. In his paper "Malware detection using statistical analysis of byte-level file content" [9] used several machine learning techniques to detect malware files. The authors claimed that their techniques can properly classify any malware regardless of its obfuscation using multi-class classification technique to detect seven classes including benign. The novelty of the authors' approach is in the ability to detect obfuscated and packed malware. The difficulty in detecting obfuscated malware lies in the obscureness of the structure of the malware file itself. The writer of the malware, intentionally, re-write the code of the file that makes it difficult to be caught by antimalware software. The total size of the content set collected for the experiment is 12,111 files (1,800 benign files and 10,311 malicious). However, only 50 files per class were used as training set. The features for the experiment generated are statistical-based features derived from byte sequence n-grams of the executables.

A static malware detection system using data mining methods proposed extraction method based on PE headers, DLLs, and API functions and methods based on Naive Bayes, J48 Decision Trees, and Support Vector Machines. The highest overall accuracy was achieved with the J48 algorithm (99% with PE header feature type and hybrid PE header and API function feature type, 99.1% with API

function feature type). [10].

Motivation of the research: The research is motivated by a serious threat called malicious executables. It is designed to damage computer system and some of them spread over network without the knowledge of the owner using the system.

Objective(s): The objective of the research is to create a static malware detection system with high accuracy rate.

Methodology: A static malware detection system using data mining techniques such as information gain, principal component analysis, and three classifiers: SVM, J48, and naïve bayes. For overcoming the lack of usual anti-virus products, methods of static analysis to extract valuable features of windows PE file was used.

Contribution to knowledge: A static malware detection system which has a detection rate of 99.6% was created.

Limitations: It is not suitable for home users because it is processor heavy

In Zero-day malware detection based on supervised learning algorithms of API call Signatures, the API functions were used for feature representation again. The best result was achieved with the Support Vector Machines algorithm with normalized poly kernel. A precision of 97.6% was achieved, with a false-positive rate of 0.025. [11].

Motivation of the research: The research is motivated by antivirus detectors being unable to detect new malwares.

Objective(s): To develop a machine learning framework using eight different classifiers to detect unknown malware and to achieve high accuracy rate

Methodology: Large data sets were used to train classifiers, and analyses the performance results of the various data mining algorithms adopted for the study using a fully automated tool developed in the research to conduct the various experimental investigations and evaluation

Contribution to knowledge: The machine learning framework developed achieved a promising result of 98.5% accuracy rate.

Limitations: API call sequence can be extracted from most device, not all which makes the algorithm limited to some devices.

The paper Survey on the usage of machine learning techniques for malware analysis [12] identify three main methods for detecting malicious software:

- Signature-based Methods.
- Heuristic-based Methods and behavior-based methods.

In addition, they [12] investigate some features for malware detection and discuss concealment techniques used by malware to evade detection. Nonetheless, the aforementioned research does not consider either dynamic or hybrid approaches.

Motivation of the research: The research was motivated by malware getting more and more challenging, given their relentless growth in complexity and volume.

Objective(s): It aims at providing an overview on the way machine learning has been used so far in the context of malware analysis in windows environments.

Methodology: For the analysis of portable executables, surveyed papers were systematized according to their objectives, what information about malware they specifically use, and what machine learning techniques they employ to process the input and produce the output.

Contribution to knowledge: It provided an overview on how machine learning algorithms can be employed in malware analysis.

Limitations: It highlighted that if the models were not properly trained it will result in running inefficient algorithms and making limited predictions.

4. Machine Learning Method

The machine learning method process consists of the following 5 stages:

i. Data intake. At first, the dataset is loaded from the file and is saved in memory.

ii. Data transformation. At this point, the data that was loaded at step 1 is transformed, cleared, and normalized to be suitable for the algorithm. Data is converted so that it lies in the same range, has the same format. At this point, feature extraction and selection, which are discussed further, are performed as well. In addition to that, the data is separated into sets – 'training set' and 'test set.' Data from the training set is used to build the model, which is later evaluated using the test set.

iii. Model Training. At this stage, a model is built using the selected algorithm.

iv. Model Testing. The model that was built or trained during step 3 is tested using the test data set, and the produced result is used for building a new model that would consider previous models, i.e., "learn" from them.

v. Model Deployment. At this stage, the best model is selected (either after the defined number of iteration or as soon as the needed result is achieved.

5. Application of Machine Learning Methods

The purpose of this section is to analyse the data and using it to train the prediction model. Figure 1 shows the sample of selected features after the feature extraction method have been applied.

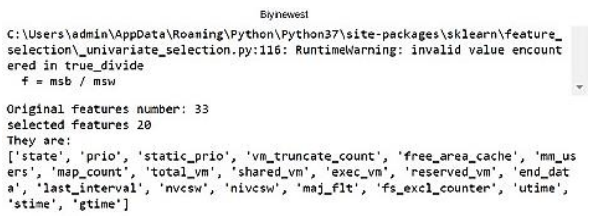

Figure 1. sample of the selected features

After the features were extracted and selected, the algorithms were applied to the data obtained. The machine learning methods applied, are K-Nearest Neighbors, Decision tree, and Random Forest. The results of the model used are shown below.

- The accuracy for the test and the training is shown in Figure 2, there is accuracy of 0.9667 on the training data and 0.9653 on the testing data.

Figure 2. Implementation of K-Nearest Neighbor Algorithm

- Figure 3 shows the accuracy of the test and training, there is accuracy of 0.9793 on the training data and 0.9779 on the testing data.

- The accuracy of the test and the training is shown in Figure 4, there is accuracy of 0.9999 on the training data and 0.9999 on the testing data.

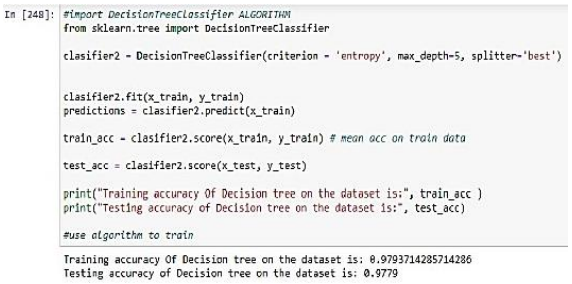

Figure 3. Implementation of Decision Tree Algorithm

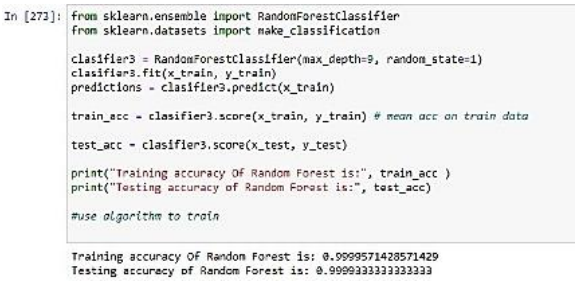

Figure 4. Implementation of Random Forest Algorithm

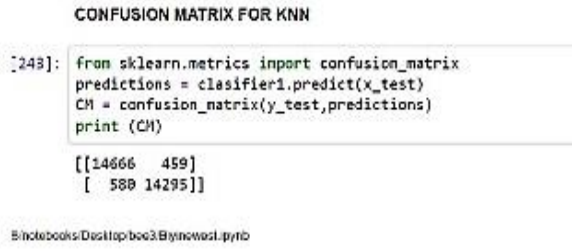

Figure 5. the confusion matrix for K-nearest Neighbor

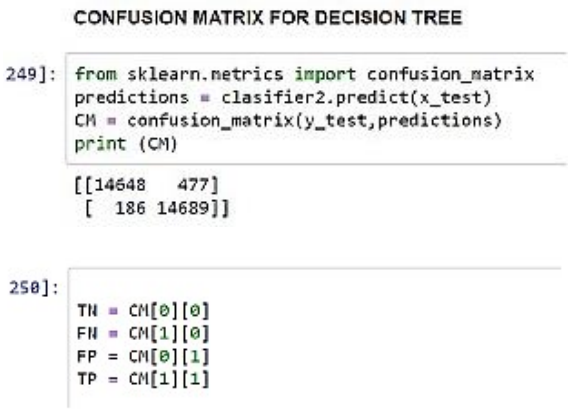

Figure 6. The confusion matrix for Decision Tree

Confusion Matrix: The confusion Matrix for K-Nearest Neighbor, Decision Tree and Random Forest

is shown in Figure 5, Figure 6, Figure 7 respectively.

CONFUSION MATRIX FOR RANDOM FOREST

```
[274]: from sklearn.metrics import confusion_matrix
       predictions = clasifier3.predict(x_test)
       CM = confusion_matrix(y_test,predictions)
       print (CM)

       [[14957     0]
        [    2 15041]]
```

Figure 7. The confusion matrix for Random Forest

7. Result of the Dataset Analysis

The overall accuracy of the algorithm is calculated below:

The overall accuracy of the algorithm is calculated using the formular $\frac{TP+TN}{TP+TN+FN+FP} * 100$

Accuracy of K Nearest Neighbor= $\frac{14295+14666}{14295+14666+580+459} * 100 = 96.53\%$

Accuracy of Decision tree= $\frac{14689+14648}{14689+14648+186+477} * 100 = 97.79\%$

Accuracy of Random Forest= $\frac{15041+14957}{15041+14957+2+0} * 100 = 99.9\%$

Table 1. Performance Evaluation Compared with other Researchers Results

S/N	Name of Researchers	Random Forest	Decision tree	K nearest neighbor
1	This work 2020	99.90%	97.79%	96.53%
2	Maqsood 2020	96.39%	100%(Overfit)	99.4%
3	Sarang *et al* 2013	99.57%	99.23%	99.06%

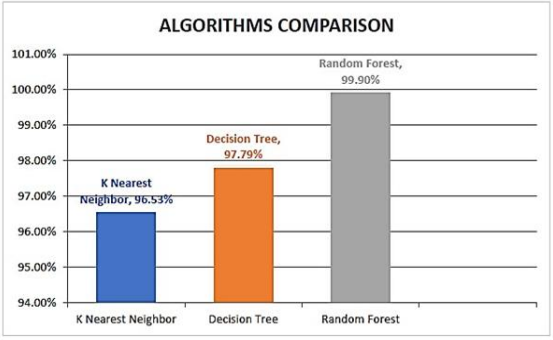

Figure 8. Algorithms performance analysis and Graphical representation of the nslkdd.csv results

Table 1 shows the performance of our research compared with other researchers which used the same machine learning algorithms (RF, DT and KNN). This work performed better than the other two [13], [14].

The Figure 8 shows that Random Forest performed better than the two other machine learning algorithms.

8. Discussion

The Figure 1 shows the original features in the dataset, but after feature selection method was applied, the features were pruned to 20. Feature selection was used in removing redundant and irrelevant features to improve the accuracy of the prediction. The Python programming language was used for performing the feature selection and applying the machine learning methods. The process was used to reduce the number of features to 20 from a total of 35 features. The method used in this research is the Filter method.

Data preprocessing technique (Data encoding and checking for missing data) was used in preparing (cleaning and organizing) the raw data to make it suitable for building and training the Machine Learning models. The reason for data preprocessing is because, it is the first step marking the initiation of the process. Typically, real-world data is incomplete, inconsistent, inaccurate (contains errors or outliers), and often lacks specific attribute values/trends. It helps to clean, format, and organize the raw data, thereby making it ready-to-go for Machine Learning models.

Figure 2 is about the result of KNN algorithm implementation which shows that the accuracy of training and testing data. KNN is a non-parametric algorithm, meaning that it does not make any assumptions about the data structure. In real-world problems, data rarely obeys the general theoretical assumptions, making non-parametric algorithms a good solution for such problems. KNN model representation is as simple as the dataset – there is no learning required, the entire training set is stored.

Figures 3 and 4 are the results of implementation of Decision Tree and Random Forest algorithm on the dataset. decision trees are data structures that have a structure of the tree. The training dataset is used for the creation of the tree, which is subsequently used for making predictions on the test data. In this algorithm, the goal is to achieve the most accurate result with the least number of decisions that must be made. The training accuracy of DT is 0.979 while the testing accuracy was 0.977. Random Forests are the collections of decision trees, producing a better prediction accuracy. That is why it is called a 'forest' – it is basically a set of decision trees. The basic idea is to grow multiple decision trees based on the independent subsets of the dataset. At each node, n variables out of the feature set are selected randomly, and the best split on these variables is found. The training accuracy for RF was 0.99 and the testing accuracy was also 0.997.

Figures 5, 6 and 7 shows the confusion matrix of the three-machine learning algorithm deployed (KNN, DT, RF). The Figure 8 is the graphical representation of the comparison performance of the three

algorithm which shows that Random Forest is the best out of the three.

9. Conclusion

Malware detection is a major issue if our computing system and its infrastructure must be kept secure in this modern age. Dataset from Kaggle.com was used for this research, the dataset which comprises of 100,000 rows and 35 columns of features was reduced to 20 by feature selection method to improve the accuracy of the algorithms. The results showed that Random Forest machine learning techniques are the best classifier to classify our data with 99.9% of accuracy.

10. References

[1] Souri, A., and Hosseini, R. (2018). A state-of-the-art survey of malware detection approaches using data mining techniques.

[2] Singhal, P., and Nataasha, R. (2015). Malware detection module using machine learning algorithms to assist in centralized security in enterprise networks. International Journal of Network Security and it's Applications, 4(1).61-67. https://arxiv.org/abs/1205.30 62 (Access Date: 23 December 2021).

[3] Jehad Ali, Rehanullah Khan, Nasir Ahmad, and Imran Maqsood. (2012). Random forests and decision trees. International Journal of Computer Science Issues (IJCSI) 9, 5.

[4] Aliyev, V. (2010). Using honeypots to study skill level of attackers based on the exploited vulnerabilities in the network(Corpus ID 107947677) [Master's thesis, Chalmers University of Technology]. https://www.semanticscholar.org/paper/usinghoneypots-to-study-skill-level-of-attackersAliyev/62cea777b89e3cc069744f5201a46d64bcafbe0. (Access Date: 21 December 2021).

[5] Shabtai, A., Kanonov, U., Elovici, Y., Glezer, C. and Weiss, Y. (2009). Andromaly: a behavioral malware detection framework for android devices.

[6] Chuvakin, A. (2003). An overview of unix rootkits. iDEFENCE inc.

[7] Lopez, W., Humberto, G., Enio, P., Erick, B., and Juan, S. (2013). Keyloggers. Florida International University.

[8] Savage, K., Peter, C., and Hon, L. (2015). The evolution of ransomware. Symantec corporation. http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/the-evolution-ofransomware.pdf (Access Date: 21 December 2021).

[9] Tabish, M., Zubair Shafiq, M., and Farooq. M. (2009). Malware detection using statistical analysis of byte-level file content. In Proceedings of the ACM SIGKDD Workshop on Cyber Security and Intelligence Informatics, pages 23–31. ACM.

[10] Baldangombo, U., Nyamjav J., and Shi-Jinn, H. (2013). A static malware detection system using data mining methods. International Journal of Artificial Intelligence and Application, 4(4), 113-126. DOI: 10.5121/IJAIA.2013.4411.

[11] Alazab, M., Sitalakshmi, V., Paul, W., and Moutaz, A. (2011). Zero-day malware detection based on supervised learning algorithms of API call signatures. In proceedings of the ninth Australasian data mining conference, 121, 171-182. Australian Computer Society. DOI: 10.5555/248 3628.2483648.

[12] Bazrafshan, Z., Hashemi, H., Fard, S.M.H. and Hamze, A. (2013).A survey on heuristic malware detection techniques Conference: Information and Knowledge Technology (IKT).

[13] Jehad Ali, Rehanullah Khan, Nasir Ahmad, and Imran Maqsood. (2012). Random forests and decision trees. International Journal of Computer Science Issues (IJCSI) 9, 5(2012), 272.

[14] Sarang Na, S. and Kwon, T. (2013). A Rolling Image based Virtual Keyboard Resilient to Spyware on Smartphones. Journal of the Korea Institute of Information Security and Cryptology. 23(6):1219-1223.