****

A close up of a logo

Description automatically generated

**Malware Detection by Machine Learning**

**Submitted by**

Bandara I B C M J

IT20624958

GitHub link:- <https://github.com/Chanaka0/-Malware-Detection-tool.git>

Contents

[Abstract 3](#_Toc146009728)

[INTRODUCTION 3](#_Toc146009729)

[MACHINE LEARNING 4](#_Toc146009730)

[MALWARE ANALYSIS 5](#_Toc146009731)

[Random Forest Classifier 5](#_Toc146009732)

[Logistic Regression Classifier 5](#_Toc146009733)

[Naive Bayes Classifier 6](#_Toc146009734)

[K-nearest Neighbors Classifier 6](#_Toc146009735)

[Neural Networks 7](#_Toc146009736)

[TECHNICAL ANALYSIS 7](#_Toc146009737)

[Detection Time 8](#_Toc146009738)

[CONCLUSION 8](#_Toc146009739)

# Abstract

Cyberattacks and the use of malware have proliferated in the modern world, affecting anything from publicly listed companies to entire countries. Malware analysis is becoming a major component of computer security event management. Untrustworthy data often come to light in forensic investigations as well as in antivirus and security monitoring systems used by enterprises. Most solutions use a combination of static and dynamic approaches, with different strategies, to discover malware. A new field of study has emerged as a result of the practical limitations of these systems.

The purpose of this research is to examine how data science can be used to detect malware and examine how machine learning methods can be used to malware analysis. Training assault detection systems is essential to building stronger defenses that can recognize new attack vectors. The results of this investigation show the variety of models that may be used to evaluate their detection performance. We demonstrate the feasibility of comparing several machine learning (ML) methods for malware evaluation through our demonstration findings.

INTRODUCTION

Users have always been at risk from harmful software, as malware is a common weapon in today's environment. It is often used for several illicit activities, such as initiating Denial of Service Attacks (DDoS), pilfering private and sensitive information, and using methods such as 0-day exploits to accelerate replication. Moreover, malicious code is often employed to generate misleading hyperlinks that direct people to hacked websites. Malware analysis is the study of various malware forms, including Trojan Horses, worms, viruses, rootkits, backdoors, and APTs, with the goal of determining the possible consequences of an infection (Afianian et al., 2018; Filiol, 2006).

Static and dynamic analysis are the two main components of malware analysis (Sikorski and Honig, 2012; Ligh et al., 2010). Using a variety of disassemblers, static analysis entails examining the contents of a malicious binary file. Analysts use methods such as image and string analysis to probe into the code, find its functions, and make inferences about its characteristics. On the other hand, malware is executed in a controlled environment as part of dynamic analysis (Willems et al., 2007) in order to watch its behavior and gain insights from it. Static analysis is faster and easier to understand, but it's important to remember that some hidden features of malware can avoid detection. To further thwart static analysis, malware developers use strategies like obfuscation and anti-disassembly techniques. When put in a sandbox for dynamic analysis, modern advanced malware might change its behavior automatically, complicating the investigation.

In recent years, malware has caused havoc in a number of industries and countries (Moubarak et al., 2017). Malware now exhibits more sophisticated behaviors and more stealth because to the deployment of novel techniques (Saad et al., 2019; Moubarak et al., 2018; Moubarak et al., 2019). Attackers have been using machine learning techniques to conceal payload and carry out harmful acts by leveraging patterns linked to artificial intelligence (AI), speech and facial recognition, and geolocation (Stoecklin, 2018). Additionally, attackers can improve target identification by utilizing machine learning (Chebbi, 2018). Compared to manual techniques, these algorithms speed up information gathering, vulnerability detection, and critical element recognition (Quinn, 2014). Furthermore, by feeding it false data structures, AI can be used to produce misleading results (James et al., 2018). Malware analysis can also make use of this technique.

In the field of machine learning (ML), understanding the traits of malicious programs is essential to differentiating between benign and malicious binaries. In order to create useful inference, this entails gathering datasets of malicious and benign binaries and extracting malware-specific features (Saxe and Sanders, 2018). Malware classification based on network activity (Boukhtouta et al., 2016), system calls (Nikolopoulos and Polenakis, 2017), APIs (Fan et al., 2015), and the detection of Android malware (Wu et al., 2016) has been the subject of numerous investigations. In order to determine if input Portable Executable (PE) files are dangerous or benign, this study examines them using a variety of machine learning methods. The datasets are assessed using models including Random Forest, Logistic Regression, Naive Bayes, Support Vector Machines, K-nearest Neighbors, and Neural Networks. In the end, a lot of testing in the real world is done to evaluate how accurate these models are.

# MACHINE LEARNING

Alan Turing made important contributions to the field of artificial intelligence throughout the 1950s, when it was first emerging (Moor, 2003). Artificial intelligence is commonly associated with computer programs that can mimic human behaviors and learn on their own. But in the field's current context, artificial intelligence primarily refers to sophisticated algorithms built to mimic human behavior. Subfields including machine learning (ML), robotics, computer vision, planning, and natural language processing (NLP) are all included in artificial intelligence (AI).

As a branch of artificial intelligence, machine learning aims to build smart machines that can replicate human behavior. When it comes to gleaning new insights from large and varied datasets, machine learning (ML) is particularly useful. Algorithms for machine learning fall into five major groups, each of which is intended to meet a distinct learning goal. When an algorithm is taught to correlate a set of labeled samples (inputs) with intended results (outputs), the process is known as supervised learning. Finding the underlying function, represented as f(x), that links inputs (x) to outputs (y) is the aim of the algorithm. Regression and classification are the two key problem types that supervised learning is good at tackling.

Conversely, unsupervised learning dispenses with the need for labels and enables the algorithm to independently discover the underlying structure in the incoming data. It creates representations of itself that could be difficult for humans to understand. Two subcategories of unsupervised learning are association and clustering. In association learning, common patterns are found in the data to form homogeneous groupings (e.g., "If X and Y occur, then event Z may follow"). Labeled and unlabeled data are combined in semi-supervised learning to improve learning efficacy (Zhu et al., 2003).

The third algorithm, reinforcement learning, is situated in between the first two. Instead than relying on labeled data analysis, it uses an experience-based incentive system. Using experiential input (successes and failures) and decision-making orientation, the process entails evaluating and reinjecting the learning algorithm to refine decision rules and identify better solutions (Littman, 1994).

Logistic regression in supervised classification only considers two possible values for (Y): negative and positive. It calculates the correlation between the likelihood of an event occurring and any possible contributing factors. Another supervised machine learning approach is Naive Bayes, which takes conditionally independent descriptors (Xi) of the variable to be predicted (Y) into account and depends on conditional probabilities.

When employed for binary classification, Support Vector Machines (SVM) look for the best hyperplane to divide two groups. According to Hastie et al. (2005), the optimal hyperplane optimizes the distance between the nearest points in each class and the separation boundary.

One flexible supervised learning method that may be used for both regression and classification tasks is K-nearest neighbors (KNN). When an observation is not included in the dataset, KNN selects the K closest examples, utilizing the complete dataset to forecast outcomes.

# Neural Networks, which have several parallel processors arranged in layers, are inspired by how the human brain learns. Like human brain networks, these networks process information from sensory input to final output through input, hidden, and output layers. The number of layers between the input and output, which is usually defined by the number of hidden nodes in the intermediate layers, determines the type of neural network.

# MALWARE ANALYSIS

The use of ML algorithms for malware detection is illustrated in this section. These algorithms are evaluated with respect to a number of malware properties, including the Import Address Table, PE headers, instructions, function calls, character strings, and compression. Python and the sklearn library were used in the construction of these methods (Saxe and Sanders, 2018).

# Random Forest Classifier

# Some issues with a decision tree are addressed in the training phase by using an automatic questioning technique. To determine if a particular sample is malicious or benign, a question is used at each node of the tree. Breiman (2001) introduced the random forest algorithm, which combines several decision trees. A random subset of samples is chosen for training, and each tree is trained using its own unique set of questions. Additionally, each subset's feature selection is randomized.

# Logistic Regression Classifier

The logistic regression classifier (Harrington, 2012) creates a boundary, either a line or a hyperplane, inside the training dataset that successfully separates dangerous software from its benign equivalents. This distinction between the two types is what the algorithm uses to determine if a new binary is potentially harmful. This border between harmful and benign software is established using the gradient descent method (Ruder, 2016). Malware is categorized according to favorable weighted features after the weighted sum of features is converted into a probability using the logistic regression classifier.

# Naive Bayes Classifier

The naive Bayes classifier determines the likelihood that a file belongs in a particular category by using a number of parameters that are taken from the training dataset. Files that exhibit a high likelihood of being clean are classified as benign ware, files with a low likelihood are classified as malware.

A graph of a detector

Description automatically generated with medium confidence

# K-nearest Neighbors Classifier

The main goal of the k-nearest neighbors method is to take into account the likelihood that a binary is malicious when the majority of its attributes closely resemble those of a malicious binary. The number of neighboring properties taken into consideration is represented by the variable 'k'. On the other hand, a binary is labeled as benign if most of the characteristics of 'k' binaries are similar to those of a benign binary. The 'k' samples' features and characteristics are compared with those of the new binary to arrive at this conclusion. A predefined threshold for sample similarity inside the feature space is met, at which point the sample is classified as benign or malicious.

This is accomplished by contrasting the characteristics and features of the new binary with those of the 'k' samples. The Euclidean distance function, which finds the lowest distance between two locations in the feature space, provides the foundation for determining feature distance. The degree of similarity between the new binary and the samples in the training set is assessed using this feature distance assessment (Wang et al., 2007; Saxe and Sanders, 2018).

The main uses of the K-Nearest Neighbor algorithm are in malware family classification and binary feature identification.

# Neural Networks

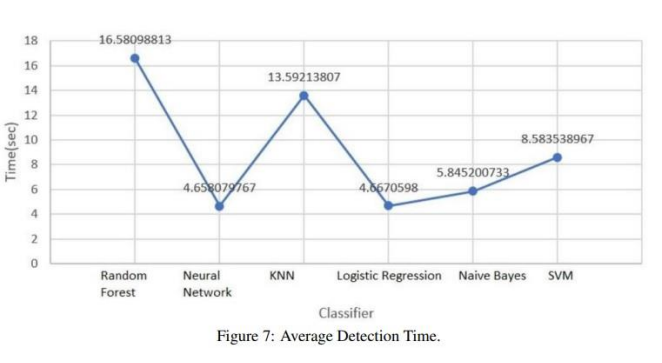
# The layers that make up the neural network's structure include an input layer, an intermediate layer, and an output layer that produces the desired outcome. 512 neurons make up the intermediate layer, which uses the Rectified Linear Unit (ReLU) activation function, as described by Saxe and Sanders (2018). According to Gan et al. (2015), the last layer uses a sigmoid function and is composed of a single neuron.

# TECHNICAL ANALYSIS

Using algorithms including Random Forest, Logistic Regression, Naive Bayes, Support Vector Machines, K-nearest Neighbors, and Neural Networks, this section analyzes malware. Saxe and Sanders provided the datasets that were used to train these models in 2018. The evaluation of these techniques takes into account elements like the Receiver Operating Characteristic Curve, detection time, and algorithmic constraints. Creating a classifier that maximizes the true positive rate and minimizes the false positive rate is the aim here.

Out of all the ROC Curves that were compared, the random forest classifier performs better. Although this classifier works well for analyzing detector plots, it can be improved even further by adding millions of more examples and growing the training dataset. It is also possible to add more criteria in order to enhance its performance even further.

# Detection Time

In order to demonstrate the efficacy of machine learning detectors, different classifiers were used to evaluate identical binaries. An overview of each classifier's detection times is shown in Figure 7. The random forest classifier demonstrated the slowest average detection time of 16.5 seconds when tested on the same new binary, whereas the neural network and logistic regression classifiers both reached the fastest detection times of 4.6 seconds.

# CONCLUSION

The importance of big data and the insights it can provide from a wide range of data sources have become increasingly apparent in the digital age. Additionally, the growing interest in data analysis has spread to a number of uses, such as the detection and defense against cyberattacks. The potential of big data and advanced analytics can now be realized thanks to recent technical breakthroughs. Machine learning algorithms are essential for identifying foreign incursions as well as internal dangers in the field of cybersecurity. For instance, they are able to spot trends in the actions of attackers who are doing reconnaissance.

It is noteworthy that the principal objective of this research is to furnish visual aids for human comprehension, so expediting the identification of significant concepts. A more thorough representation of common scenarios is produced by combining information from many sources, including system log files, historical IP address data, honeypots, and system and user behaviors. In order to identify malicious activity, this thorough technique examines a variety of sources and patterns.

In addition, machine learning is used in a variety of penetration testing use cases, including attack attribution and detection. This paper presents research that demonstrates the application of various machine learning algorithms for malware detection. A thorough description of each algorithm's malware detection procedure has been provided, and numerous methods have been put into practice, trained, and tested. Furthermore, each classifier's Receiver Operating Characteristic (ROC) Curve is displayed, emphasizing the disparities in algorithmic performance.

Although some algorithms might have average detection times that are longer than those of others, this study's classifier evaluation highlights the random forest classifier's good performance in contrast. Our long-term goals include investigating and improving hybrid trail models and ensemble learning methods for malware detection. By adding more parameters and growing the training dataset, these algorithms can be improved. In addition, we intend to use many analysis modalities, including machine learning, dynamic, and static approaches, in order to provide an all-encompassing malware identification tool.

# REFERENCES

1. <https://www.udemy.com/course/machinelearning/?matchtype=p&msclkid=cf7e03fdab791b9959e160a7d35f9bac&utm_campaign=BGLongTail_la.EN_cc.ROW&utm_content=deal4584&utm_medium=udemyads&utm_source=bing&utm_term=_._ag_1219358808347883_._ad__._kw_%2BMachine+%2BLearning+%2Bguide_._de_c_._dm__._pl__._ti_kwd76210238096187_._li_147906_._pd__._>
2. <https://www.eclecticiq.com/platform?hsa_mt=p&hsa_kw=threat%20analysis&hsa_tgt=kwd78752980884819&hsa_src=o&hsa_acc=3947891926&hsa_grp=1260041052850304&hsa_cam=12863663931&hsa_net=adwords&hsa_ad=&hsa_ver=3&utm_source=bing&utm_medium=cpc&utm_campaign=1.%20InformationalG12863663931&utm_term=threat%20analysis&utm_content=1.3%20Use%20Case%20-%20CTI>
3. <https://towardsdatascience.com/random-forest-classifier-projectpredicting-online-news-popularity-e99c79daa33>
4. <https://www.researchgate.net/publication/260820980_Logistic_Regression_Classification_for_Uncertain_Data>
5. <https://ee104.stanford.edu/lectures/neural.pdf>
6. Moubarak, J., Chamoun, M., and Filiol, E. (2017). Comparative study of recent mea malware phylogeny. In 2017 2nd International Conference on Computer and Communication Systems (ICCCS), pages 16–20. IEEE.