IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

# Supervised Machine Learning Algorithms for the Detection of Malware Activities

[1]Dr. Manjunatha S, [2]Dr. Preethi S, [3]Dr.Bharani B R, [4]Prof. Vijayalakshmi R Y

[1]Associate Professor, [2] Professor, [3]Associate Professor, [4]Assistant Professor
[1]Computer Science and Engineering, [2,3,4]Information Science and Engineering,
[1,2,3,4]Cambridge Institute of Technology, Bangalore, India.

***Abstract:*** Malware Detection is a significant part of endpoint security including workstations, servers, cloud instances, and mobile devices. Malware Detection is used to detect and identify malicious activities caused by malware. With the increase in the variety of malware activities on different files online and offline, It's Important for Data Security, Privacy and protection. So We will use Machine Learning and its algorithm to see the accuracy and prediction on Malware Datasets. In this Project we will use many different algorithms for analysing and studying the Malware in Dataset.

***IndexTerms* - Malware detection, viruses, machine learning**

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 analysing 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. Current studies show that script-kiddies are generating more and more attacks or are automated. To date, attacks on commercial and government organisations, 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 multinational law enforcement agencies. For single users and organisations, malware defence 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. Mobile phones have become increasingly important tools in people's daily life, such as mobile payment, instant messaging, online shopping, etc., but the security problem of mobile phones is becoming more and more serious. Due to the open source nature of the Android platform, it is very easy and profitable to write malware using the vulnerabilities and security defects of the Android system. This is the main reason for the rapid increase in the number of malware on the Android system.

**LITERATURE SURVEY**

Christodorescu et al., 2005 [3] described a malware instance as a program whose objective is malevolent. McGraw and Morrisett,2000 defined malicious code as "any code added, changed, or removed from a software system in order to intentionally cause harm or subvert the intended function of the system." The description given by (Vasudevan and Yerraballi, 2006) which described malware as "a generic term that encompasses viruses, trojans, spywares and other intrusive code." [4] (Aycock, 2006) defined malware as "software whose intent is malicious, or whose effect is malicious". [5] The term "malware" here is being used as the generic name for the class of code that is malicious, including viruses, trojans, worms, and spyware. Malware authors use generators, incorporate libraries, and borrow code from others—there exists a robust network for exchange, and some malware authors take time to read and understand prior approaches by (Arief & Besnard ,2003.) [6] (Fred Cohen's) original definition of a computer virus as of 1983 was: "a program that can 'infect' other programs by modifying them to include a possibly evolved copy of itself." He updated this definition a year later in 1984 in his paper entitled: "Computer Viruses – Theories and Experiments". [7] According to BBC News online, 2004 malware is a general term for a piece of software inserted into an information system to cause harm to that system or other systems, or to subvert them for use other than that intended by their owners.

## METHODOLOGIES

Machine learning can easily identify the malware in the data and datasets. Different types of machine learning algorithms are applied such as:

- SVM
- Random forest
- XG boost

### Random forest

A random forest is a machine learning technique that's used to solve regression and classification problems. It utilizes ensemble learning, which is a technique that combines many classifiers to provide solutions to complex problems.
A random forest algorithm consists of many decision trees. The 'forest' generated by the random forest algorithm is trained through bagging or bootstrap aggregating. Bagging is an ensemble meta-algorithm that improves the accuracy of machine learning algorithms.

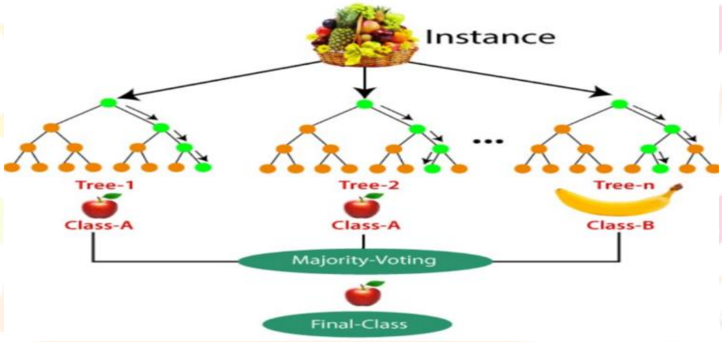


**Fig 1: Random Forest**

### XG boost

XGBoost or extreme gradient boosting is one of the well-known gradient boosting techniques(ensemble) having enhanced performance and speed in tree-based (sequential decision trees) machine learning algorithms. It is the most common algorithm used for applied machine learning in competitions and has gained popularity through winning solutions in structured and tabular data.


**Fig 2: XG Boost**

### SVM

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.

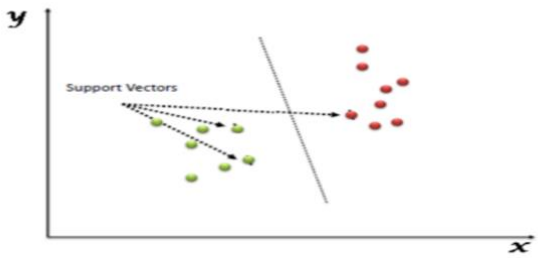


**Fig 3: SVM**

This section depicts how ML algorithms are evoked to detect malware. The evaluation of the algorithms considered multiple malware features including PE headers, instructions, calls, strings, compression and the Import Address Table. The implementation was based on Python and sklearn (Saxe and Sanders, 2018).

**Random Forest Classifier**

A decision tree solves problems by automatically generating interrogation while training samples. For each node of the tree, a question is employed to decide whether the sample is a malware or a benignware. The random forest algorithm (Breiman, 2001) combine multiple decision tree where each tree is trained using different questions. Each tree was trained using a random chosen partial set of samples and the features of each sets are randomly selected. hashed features hasher.transform ([string features]) The detection of a sample is performed on every tree and the algorithm decides on the maliciousness of the binary founded on the response of the majority of the trees.

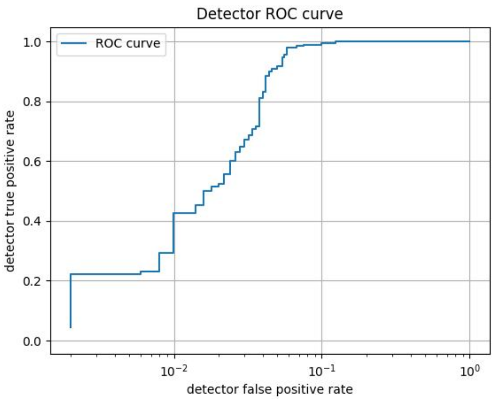


**Fig 4: Random forest classifier performance graph**

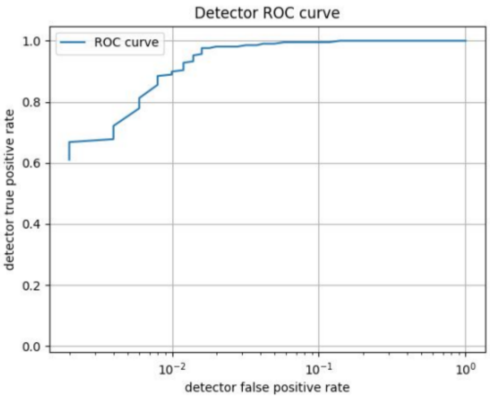


**Fig 5: Support vector machines classifier performance**

**Support Vector Machines Classifier**

The support vector machine classifier will also draw a hyperplane that splits malware from benignware in the training dataset. The decision of being clean or suspicious depends on its location compared to the hyperplane (Chebbi, 2018).

**Table 1: Classifiers' Performance**

| Classifier | FPR | TPR |
|---|---|---|
| Random forest classifier | 0.01 | 92% |
| Logistic regression classifier | 0.01 | 40% |
| Naive baise classifier | 0.021 | 65% |
| Support vector machines classifier | 0.01 | 40% |
| K-nearest neighbors classifier | 0.01 | 59% |
| Neural network classifier | 0.01 | 82% |

## TECHNICAL ANALYSIS

This section analyses malware leveraging Random Forest, Logistic Regression, Naive Bayes, Support Vector Machines, K-nearest neighbors and Neural Networks algorithms. These models were trained on datasets from (Saxe and Sanders, 2018). The evaluation between these algorithms includes the Receiver Operating Characteristic Curve, the detection time and the limitation of each algorithm.

### Receiver Operating Characteristic Curve

The Receiver Operating Characteristic Curve (ROC Curve) permits to predict the correctness of machine learning algorithms (Bradley, 1997). It consists of a plot visualizing the algorithm true positive rate (TPR) versus its false positive rate (FPR).

$$TPR = \frac{True\ Positives}{True\ Positives + False\ Negatives} \qquad FPR = \frac{False\ Positives}{False\ Positives + True\ Negatives}$$

Predictive rates are divided in four classes:

- True Positive (TP): The binary is correctly predicted as being malicious.
- True Negative (TN): The binary is correctly predicted as being benign.
- False Positive (FP): The binary is incorrectly predicted as being malicious.
- False Negative (FN): The binary is incorrectly predicted as being benign.

A good classifier will try therefore to maximize the true positive rate and to minimize the false positive rate. Comparing the several ROC Curves (Table 1), the performance of the random forest classifier is the best among the other algorithms. Besides, comparing the plots of the detectors, the random forest classifier performs well, noting that the execution can be enhanced when scaling the training dataset to a bigger amount of test data adding millions of samples. Others parameters can be added as well.

### Detection Time

To highlight the performance of ML detectors, the same binaries were tested using the different classifiers. Figure 7 summarizes the detection time of each classifier. For a same new binary to test, the neural network and logistic regression classifier achieved the fastest detection rate (4.6 secondes) and the random forest classifier the slowest average (16.5 secondes).

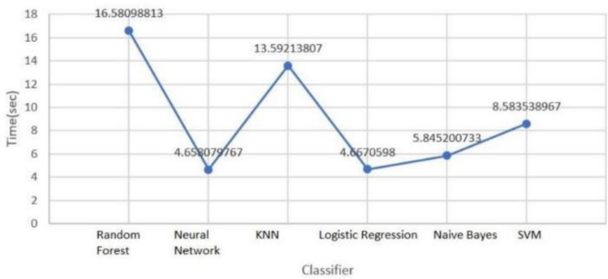


**Fig 6: Average**

## CONCLUSION

The information age has recently discovered the value of big data and information that can hide in disparate, large data sources. The current interest in data has also spread across multiple applications to detect and prevent attacks. New technologies permit nowadays an advanced analytics approach leveraging big data. In cybersecurity, machine learning algorithms can be used to detect external intrusions, for example by identifying patterns in the behavior of attackers performing reconnaissance, but also to detect internal risks. The analysis simply aims to provide visualization so that human interaction can be applied to infer ideas. By combining data from system log files, historical data on IP addresses, honeypots, system and user behaviors, etc. a more comprehensive overview of a normal situation is conceived. The wit is to analyze multiple sources and patterns to signal unwanted behavior. Furthermore, machine learning is used for attack detection and attribution. Besides, several use cases of machine learning are employed for penetration testing. The work done in this paper proves that different approaches can be leveraged to detect malware using machine learning. Several algorithms have been implemented, trained and tested. For each algorithm, the methodology of detecting malware have been abridged in details. Moreover, the ROC Curve of each classifier has been illustrated showing that some algorithms perform better than others. This study and classifiers' evaluation show that random forest operates satisfactorily comparing to other algorithms even that the average detection time is not the lowest. Our future plans consist in studying and enhancing the detection of malware using hybrid training model and ensemble learning. These algorithms can be built also leveraging other parameters and training data. In addition, in a next step we envisage to associate multiple analysis techniques to detect malware. For a complete detection mechanism, we plan to combine static, dynamic and machine learning techniques to analyse malware.

## REFERENCES

[1] Afianian, A., Niksefat, S., Sadeghiyan, B., and Baptiste, D. (2018). Malware dynamic analysis evasion techniques: A survey. arXiv preprint arXiv:1811.01190.

[2] Boukhtouta, A., Mokhov, S. A., Lakhdari, N.-E., Debbabi, M., and Paquet, J. (2016). Network malware classification comparison using dpi and flow packet headers. Journal of Computer Virology and Hacking Techniques, 12(2):69–100.

[3] Bradley, A. P. (1997). The use of the area under the roc curve in the evaluation of machine learning algorithms. Pattern recognition, 30(7):1145–1159.

[4] Breiman, L. (2001). Random forests. Machine learning, 45(1):5–32.

[5] Chebbi, C. (2018). Mastering Machine Learning for Penetration Testing. Packt Publishing.

[6] Fan, C.-I., Hsiao, H.-W., Chou, C.-H., and Tseng, Y.-F. (2015). Malware detection systems based on api log data mining. In 2015 IEEE 39th annual computer software and applications conference, volume 3, pages 255–260. IEEE.

[7] Filiol, E. (2006). Computer viruses: from theory to applications. Springer Science & Business Media.

[8] Gan, Z., Henao, R., Carlson, D., and Carin, L. (2015). Learning deep sigmoid belief networks with data augmentation. In Artificial Intelligence and Statistics, pages 268–276.

[9] Harrington, P. (2012). Machine learning in action. Manning Publications Co. ForSE 2020 - 4th International Workshop on FORmal methods for Security Engineering 850

[10] Hastie, T., Tibshirani, R., Friedman, J., and Franklin, J. (2005). The elements of statistical learning: data mining, inference and prediction. The Mathematical Intelligencer, 27(2):83–85.

[11] James, J., Hou, Y., and Li, V. O. (2018). Online false data injection attack detection with wavelet transform and deep neural networks. IEEE Transactions on Industrial Informatics, 14(7):3271–3280.

[12] Ligh, M., Adair, S., Hartstein, B., and Richard, M. (2010). Malware analyst's cookbook and DVD: tools and techniques for fighting malicious code. Wiley Publishing. Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. In Machine learning proceedings 1994, pages 157–163. Elsevier.

[13] Moor, J. (2003). The Turing test: the elusive standard of artificial intelligence, volume 30. Springer Science & Business Media.

[14] 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.

[15] Moubarak, J., Chamoun, M., and Filiol, E. (2018). Developing a k-ary malware using blockchain. In NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium, pages 1–4. IEEE. Moubarak, J., Chamoun, M., and Filiol, E. (2019). Hiding malware on distributed storage. In 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), pages 720–725. IEEE.

[16] Nikolopoulos, S. D. and Polenakis, I. (2017). A graphbased model for malware detection and classification using system-call groups. Journal of Computer Virology and Hacking Techniques, 13(1):29–46.

[17] Quinn, M. J. (2014). Ethics for the information age. Pearson Boston, MA.

[18] Ruder, S. (2016). An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747.

[19] Russell, S. J. and Norvig, P. (2016). Artificial intelligence: a modern approach. Malaysia; Pearson Education Limited,.

[20] Saad, S., Briguglio, W., and Elmiligi, H. (2019). The curious case of machine learning in malware detection. arXiv preprint arXiv:1905.07573.

[21] Saxe, J. and Sanders, H. (2018). Malware Data Science. No Startch Press.

[22] Sikorski, M. and Honig, A. (2012). Practical malware analysis: the hands-on guide to dissecting malicious software. no starch press.

[23] Stoecklin, M. P. (2018). Deeplocker: How ai can power a stealthy new breed of malware. Security Intelligence, 8.

[24] Wang, J., Neskovic, P., and Cooper, L. N. (2007). Improving nearest neighbor rule with a simple adaptive distance measure. Pattern Recognition Letters, 28(2):207–213.

[25] Willems, C., Holz, T., and Freiling, F. (2007). Toward automated dynamic malware analysis using cwsandbox. IEEE Security & Privacy, 5(2):32–39.

[26] Wu, S., Wang, P., Li, X., and Zhang, Y. (2016). Effective detection of android malware based on the usage of data flow apis and machine learning. Information and Software Technology, 75:17–25.

[27] Zhu, X., Ghahramani, Z., and Lafferty, J. D. (2003). Semisupervised learning using gaussian fields and harmonic functions. In Proceedings of the 20th International conference on Machine learning (ICML-03), pages 912–919. Comparing Machine Learning Techniques for Malware Detection