Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 45 No. 2 (2024)

Machine Learning Approaches for Malware Detection: An Analytical Overview

[1]Abinash Patra, [2]Kanishk Rao,[3]Ashish Deb,[4]Anurikt Kumar,[5]Siddhant Arya,[6]Gargi Sharma

Lovely Professional University
Kantabanji, Odisha, India

Abstract — The constantly changing character of malware makes "difficult to resolve using conventional signature-based detection methods. This paper discusses possible applications of Machine Learning (ML) algorithms to enhance malware detection precision. A comparative analysis is conducted on five prominent ML algorithms: Support Vector Machines (SVM), Decision Trees (DT), Neural Networks (NN), Random Forests (RF), and Logistic Regression (LR). Performance metrics, e.g., F1 score, TPR, FPR, and a Confusion Matrix analysis are used for picking up the most suitable algorithm for a given dataset. This research, however, highlights the need for achieving an equilibrium between accurate malware detection and lowering the rate of false positives. Through the evaluation of the plusses and minuses of each ML algorithm, steps are taken towards the improvement of detection performance, especially that of the variants of elusive and polymorphic malwares. These results will contribute to the development of ML-based cybersecurity strategies that may be effective against constantly emerging malware threats in the digital space.

Index Terms — Malware, Detection techniques, Signature based, Heuristic based, Behavioural Analysis, Machine Learning, Cyber Security.

1. Introduction

In the contemporary digital sphere, the widespread circulation of malicious applications/software, mainly the well-known – malware poses a severe cyber threat. Malware is a broad term and includes all the bad programs which are specially designed to get into systems, disrupt their functioning or eventually damage them. Given the rapid and cutting-edge development of cyber threats, effective techniques are essential to combat them. Therefore, considerable techniques for this purpose are essential for defeating cyberthreats.

Malware analysis is the way of analyzing and understanding malicious software to achieve purpose of identifying malware' characteristics, behavioral pattern and potential impact. This reactive stance can be assessed by discovering countermeasures, search for malicious behaviours and hardening the defences. Throughout the years the malware has been researched and examined from various directions, including methods and methodologies such as dynamic analysis, static analysis reverse engineering and behavioral analysis.

Static analysis consists of searching the malware code and its structure without the execution of the same, subsequently revealing what the malware is capable of and possibly the best attack approach thereof. In contrary, non-dynamic and static analysis exposes malware to fewer environments with no chances for changing necessary systems states. On the other hand, a dynamic or real-time analysis takes place inside a sandbox, where malware is run in order to observe how it behaves and how it interacts with the system. Activities evaluation is a technique which monitors actions of malware while it is being executed and which is used to detect and classify malicious behavioral patterns and their signatures. Along with this, the dismantling of bits and pieces of this malware at large and the consequent understanding of internal workings is also used in order to come up with the possible weak points of malware and develop effective cyber defences.

Though the conventional literature of the malware study and the traditional processing methods sometimes have been useful in curbing cybercrime, the swift evolution of the malware and the massive intensify in the number of the malicious samples are prominent challenges for manual analysis approaches. Finally, there are the emergence of ‘big data’ phenomena that result in a growing demand to apply artificial intelligence and machine learning for malware analysis dealing with ever evolving threats. In the past years, researchers have developed an efficient technique called deep learning, which combines the learning model having similarities with the neural networks of the human brain, for the purpose of malware analysis.

Deep learning approaches undergo the process of self-learning and autonomously decode complex features from overwhelming data volumes, hence making it easier for them to identify faint details and aberrant patterns that focus also on malware behavior. Using deep learning solutions, scientists together with analysts will be able of constructing robust schemes for malware detection and classification, which do not perplex an already unseen symbol with high accuracy and speed.

The objective of this research is to assess how different techniques and methods of malware analysis are applied, pointing out that deep learning acts as an adjunct, bolstering malware detection and mitigation procedures. By working through the fundamental principles that deep learning holds in malware analysis, this study aims to contribute knowledge for sophisticated technology to overcome the continuous changing nature of cybersecurity.

2. Literature review

A. Machine Learning-Based Approaches:

Machine learning algorithms are being used widely for malware detection due to the fact that they are data-driven and can recognize behavioural characteristic of malware [1]. A study by Supriya et al. In 2019 and Al-Janabi & Altamimi in 2020 focused on SVM, KNN, RF, and MLP as classification algorithms to be used in detecting malicious files based on extractable features [1], [2].

B. Deep Learning-Based Approaches:

Deep learning, with the ability to automatically identify even the most intricate patterns from raw unprocessed data has been intensively used for malware detections [3]. Sreekumari's research (2020) targets deep learning architectures like Convolutional Neural Networks (cnns), Recurrent Neural Networks (rnns), and LSTM networks to unravel the malware samples created as opcode sequence representations or API call sequences [3]. Furthermore, other investigation concentrates on designing feature engineering and representation methods for enhancing performance of DL models in malware detection as demonstrated by works of Zhang et al. (2020) and Catak et al. (2020) [4], [5].

C. Hybrid Approaches:

Some scientists have designed hybrid approaches consisting of wiring up several methods, which would make malware detection more effective as a whole [6]. Moubarak & Feghali assess the capabilities of different ML and deep learning approaches, showing the strengths and weaknesses of each methodology [6]. Likewise, Altamimi and Al-Janabi conducted an experiment about exploiting static, dynamic analysis techniques that amount to the overall malware detection [7].

D. Dynamic Analysis Techniques:

Dynamic approach includes dynamically analyzing the malware in a well-controlled environment like a sandbox in order to see their activities and attempt to clamp down on the malicious deeds [8]. Authors Aslan and Samet, in their research (2020), and Tibra and Alqudah, in their work (2018) suggest that dynamic analysis techniques with the help of sandboxing and behaviour monitoring tools can be used to perceive the presence of malware by segmenting their operating features [8], [9].

E. Feature Engineering and Selection

Feature engineering as the machine learning part, which is also known as converting raw data into forms of representation that can be interpreted easily and analysed is of the utmost significance in malware detection [10]. The former work by Ranveer et al. And the latter work by Rabbani et al. Investigated diverse “feature extracting methods" which are opcode frequency, API calls, or the handle tables, to figure out the main features those malware samples hold [10], [11]

Table 1
Synopsis of the above literature review.

Author (Year)	Techniques	Accuracy
Catak et al. (2020)	Windows API call, LSTM(RNN), TF-IDF	83.5% - 98.5%
Zhang et al. (2020)	Gated-CNN, Bi-LSTM	98.80%
Rabbani et al. (2020)	Network features	96.50%
Namavar et al. (2020)	Behavioural features	99.65%
Yucel et al. (2020)	Memory Images	99.50%

3. Methodologies

A. Dataset

One of the most important aspects of this work is the dataset **(malwaredata.csv)** that has been used which is from the third chapter of the book *"Mastering machine learning for penetration testing,"* which contains various malware and benign information. This dataset contains 41,323 instances of benign files and 96,724 instances of malware files ranging over 56 features.

Since machine learning model runs on numerical values, the dataset contained some string values such as name of the file and md5 hash values. Before using each model all the string values have been cleared.

B. Algorithms used:

i. Neural Network:

Neural Network (NN) models operates on patterns similar to the biological structure of human brain which consist of nodes organized in layers [12]. These layers receive data and store patterns by decoding it. Think of applying one NN on many cat-images in the training process. The neural network has means to recover connection between nodes and in new images recognizes cats [12].

This sequential learning process is called backpropagation, which means error functions are fed back to the network through which it can continuously improve self-learning capability to identify patterns. The biggest advantage of neural networks is that they have been successfully used for the tasks that humans find the most complex, such as image recognition and speech recognition, because of their capacity to learn overly complex relationships from huge amounts of data [12].

ii. DT (Decision Tree):

A decision tree a supervised learning algorithm, is traditionally used for both regression and classification tasks [13] It operates by iteratively dividing the dataset into smaller subsets, focusing on reducing homogeneity based on the value of dimensionality reduction that provides the highest information gain [13].

Decision tree cannot be defined by a fixed formula, but it involves iteratively choosing features that will disclose the most effective split (based on information gain/similar splitting criteria) [14]. Through this way, the decisions trees are able to learn and predict decision-making process based on the known data in a clear and understandable way [13].

iii. SVM (Support Vector Machine):

A support vector machine (SVM) is a type of supervised learning algorithm used in machine learning to solve classification and regression tasks; svms perform particularly well with binary classification issues that demand the elements of a dataset to be classified into two sets. Based on the principle that the model divides data into hyper levels by classifying the data points belonging to different classes which, in turn, increases the space between classes and decreases error. [15]

Svms seek to maximize the margin that separates these support vectors while minimizing the errors of classification at the same time [16]. The whole approach is based on building a convex optimization problem where a feasible solution can be derived.[17]

iv. RF (Random Forest):

RF deals with a collective learning technique which involves many decision and subset trees [16]. It produces a plethora of decision trees in its training phase to arrive at the final output by combining their predictions and probably adopting the mode (most frequent prediction) or mean (average prediction) for the classification and regression tasks, respectively.[16].

Random forests do not have specified formulas but the power of the RHF is what combines predictions from many anomaly trees (in some cases they make use of techniques such as Boosting, where each tree is trained with a randomly chosen subset of data, and feature randomization, when a randomly selected feature is used at each split).

v. LR (Logistic Regression):

Logistic regression is used for the purposes of modelling the instances which represent either category [5]. Different from the decision tree and the SVM classifiers used in the prediction of class labels, logistic regression determines the probability of an instance as belonging to a particular class [18].
This method is applied through a function that is characteristic to an S-shaped curve (sigmoid function) that transforms input data into a probability value which ranges from 0 (not belonging to the class) to 1 (belonging to the class). This outcome gives great information which helps to form the degree of chance that an event belongs to a particular class [19], [20].

C. Some evaluation matrices

i. F1 Score:

F1 score is a metric which demonstrates balance between precision and recall that can be used positively and yields the most comprehensive view of the classifier's performance therefore, goal can be achieved [23]. Usually, it (F1 statistic) is represented as the harmonic mean of precision and recall, and the higher the score of F1, the higher quality the performance has [23].
*F1 Score = 2 * (Precision * Recall) / (Precision + Recall)*

ii. Confusion Matrix:

The confusion matrix is a visual representation that determines the accuracy of classifiers and thus, is used to interpret different classification algorithms [24]. It is a table where rows mean the existing classes and columns the possibilities [24].

iii. True Positive Rate (TPR):

True positive rate (TPR), which is also called recall or sensitivity, is a measure for correct answers. It determines the number of positive ones which the classifier has correctly identified of all positive cases [25].
TPR = True Positives / (True Positives + False Negatives)

iv. False Positives:

False positives may refer to the scenarios where the classifier misidentifies negative outcome as a positive one [30]. Plainly stated, it could be considered as a false alarm (e.g., categorizing an innocent file as malware) when it is not supposed to be so [21] False positives, which is among the confusion matrix elements beside recall, contributes to precision calculation. The formula for the rate of false positives is:
False Positives / (False Positives + True Negatives)

These metrics are critical for understanding the performance of classifiers and especially in a binary classification task. They help to evaluate the model's balancing between the sensitivity it has towards the positive instances and the values that correctly identified the negative examples.

4. Result and analysis

The dataset was split into an 80:20 ratio for the purpose of training and testing. Using aforementioned models the following table is derived:

Table 2: Result of every model used:

Algorithm	**Accuracy**	**F1 score**	**TPR**	**FPR**
Neural Network	0.968	0.948	0.968	0.031
Decision Tree	**0.992**	**0.987**	**0.987**	**0.005**
Support Vector Machine	0.697	0.0002	0.0001	0.00
Random Forest	0.983	0.973	0.966	0.008
Logistic Regression	0.956	0.923	0.871	0.006

By examining the above table, the decision tree model is best suited for determining different malwares and benign files. It has achieved 99.2% accuracy in determining the differences between a malware and benign file with a high TPR and Low FPR with satisfactory F1 Score.

Detailed analysis of the matrices achieved using decision tree are given next.

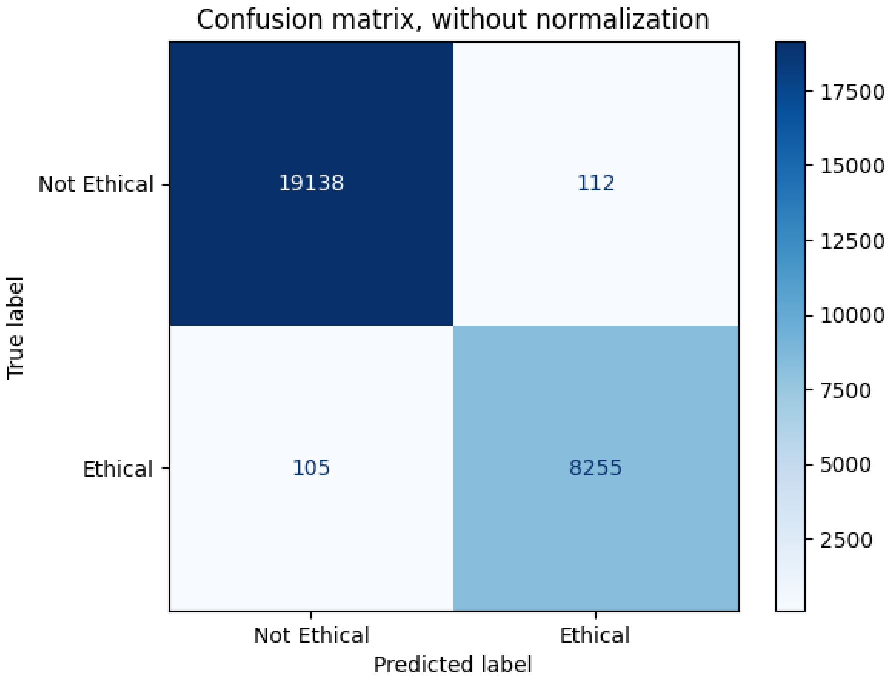

Fig1. Confusion matrix of DT

The Decision Tree model's confusion matrix (Fig 1) shows a respectably low false positive rate (FPR) of 0.38% and a high true positive rate (TPR) of 99.62%. This implies that the model minimizes false alarms while accurately identifying malware instances.

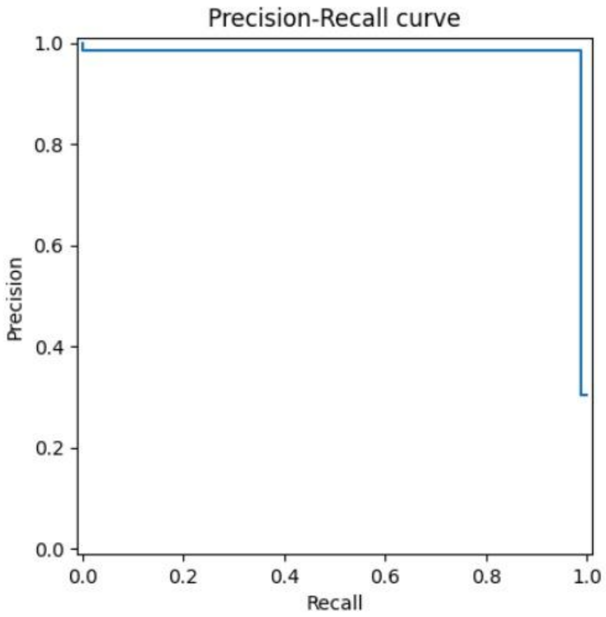

Fig2. Precision recall curve

The PR curve shows a good trade-off between precision and recall. At high precision (~1.0) recall remains acceptable. This balance aligns well with the classification task's priorities.

5. Conclusion

In cyber security industry, ML has been successful in redefining malware detection with more speed, adaptability and efficiency which makes it necessary to tackle the rampant threats. This work has rigorously analysed ML methods, algorithms, and possible future developments, through which a complete knowledge of the essential features of this field can be achieved. By leveraging the ability of ML and confronting the sophisticated nature of its challenges, the path for a more secure digital environment is cleared.

The interaction among security analysts, ML specialists and cybersecurity practitioners is the most fundamental factor for strengthening the defences against the complex malware. The way to create one's robust cybersecurity mechanisms involves recurrent cycle of learning, innovation and implementation. The integration of the ML-driven analysis and real-time threat intelligence gives an opportunity to anticipate and neutralize emerging threats prior to their occurrence.

In the near future the convergence of ML with other advanced technologies e.g. Blockchain, Edge computing and Quantum computing is bound to create a strong base for cyber resilience. The adoption of multi-disciplinary collaborations and cultivating a culture of inventiveness will hence play a critical role in setting up a dynamic and resilient cybersecurity framework.

"In the era of fast-growing digital ecosystems, we remain committed to using the full potential of ML technology and constantly improving it in order to secure digital data and personal information. Through being alert, proactive, and working together, we shall create an environment where cyber security becomes the bedrock of our digital era."

References

[1] K. Shaukat, S. Luo, V. Varadharajan, I. A. Hameed, and M. Xu, "A Survey on Machine Learning Techniques for Cyber Security in the Last Decade," *IEEE Access*, vol. 8, pp. 222310–222354, 2020, doi: 10.1109/ACCESS.2020.3041951.

[2] Y. Supriya, G. Kumar, D. Sowjanya, D. Yadav, and D. L. Kameshwari, "Malware detection techniques: A survey," in *PDGC 2020 - 2020 6th International Conference on Parallel, Distributed and Grid Computing*,

Institute of Electrical and Electronics Engineers Inc., Nov. 2020, pp. 25–30. Doi: 10.1109/PDGC50313.2020.9315764.

[3] P. Sreekumari, "Malware Detection Techniques Based on Deep Learning," in *Proceedings - 2020 IEEE 6th Intl Conference on Big Data Security on Cloud, bigdatasecurity 2020, 2020 IEEE Intl Conference on High Performance and Smart Computing, HPSC 2020 and 2020 IEEE Intl Conference on Intelligent Data and Security, IDS 2020*, Institute of Electrical and Electronics Engineers Inc., May 2020, pp. 65–70. Doi: 10.1109/bigdatasecurity-HPSC-IDS49724.2020.00023.

[4] Z. Zhang, P. Qi, and W. Wang, "Dynamic Malware Analysis with Feature Engineering and Feature Learning," Jul. 2019, [Online]. Available: http://arxiv.org/abs/1907.07352

[5] F. O. Catak, A. F. Yazi, O. Elezaj, and J. Ahmed, "Deep learning based Sequential model for malware analysis using Windows exe API Calls," *peerj Comput Sci*, vol. 6, pp. 1–23, 2020, doi: 10.7717/PEERJ-CS.285.

[6] J. Moubarak and T. Feghali, "Comparing Machine Learning Techniques for Malware Detection," in *International Conference on Information Systems Security and Privacy*, Science and Technology Publications, Lda, 2020, pp. 844–851. Doi: 10.5220/0009373708440851.

[7] M. Al-Janabi and A. M. Altamimi, "A comparative analysis of machine learning techniques for classification and detection of malware," in *Proceedings - 2020 21st International Arab Conference on Information Technology, ACIT 2020*, Institute of Electrical and Electronics Engineers Inc., Nov. 2020. Doi: 10.1109/ACIT50332.2020.9300081.

[8] O. Aslan and R. Samet, "A Comprehensive Review on Malware Detection Approaches," *IEEE Access*, vol. 8. Institute of Electrical and Electronics Engineers Inc., pp. 6249–6271, 2020. Doi: 10.1109/ACCESS.2019.2963724.

[9] T. Alsmadi and N. Alqudah, "A Survey on malware detection techniques," in *2021 International Conference on Information Technology, ICIT 2021 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., Jul. 2021, pp. 371–376. Doi: 10.1109/ICIT52682.2021.9491765.

[10] S. Ranveer and S. Hiray, "Comparative Analysis of Feature Extraction Methods of Malware Detection."

[11] B. Anderson, D. Quist, J. Neil, C. Storlie, and T. Lane, "Graph-based malware detection using dynamic analysis," *Journal in Computer Virology*, vol. 7, no. 4, pp. 247–258, Nov. 2011, doi: 10.1007/s11416-011-0152-x.

[12] M. S. C. Thomas and J. L. Mcclelland, "Connectionist Models of Cognition," in *The Cambridge Handbook of Computational Psychology*, Cambridge University Press, 2012, pp. 23–58. Doi: 10.1017/cbo9780511816772.005.

[13] Breiman Leo, Friedman Jerome, Olshen Richard, and Stone Charles, *Classification and Regression Trees*. 1984.

[14] J. R. Quinlan, "Induction of Decision Trees," 1986.

[15] C. Cortes, V. Vapnik, and L. Saitta, "Support-Vector Networks Editor," Kluwer Academic Publishers, 1995.

[16] L. Breiman, "Random Forests," 2001.

[17] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A Training Algorithm for Optimal Margin Classifiers."

[18] A. Liaw and M. Wiener, "Classification and Regression by randomforest," 2001. [Online]. Available: https://www.researchgate.net/publication/228451484

[19] G. (Gareth M. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning : with applications in R*.

[20] T. Hastie, R. Tibshirani, and J. Friedman, "Springer Series in Statistics The Elements of Statistical Learning Data Mining, Inference, and Prediction."

[21] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit Lett*, vol. 27, no. 8, pp. 861–874, Jun. 2006, doi: 10.1016/j.patrec.2005.10.010.

[22] Szaszko Matt, "Recall vs Precision in an AI product or feature," Medium.

[23] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf Process Manag*, vol. 45, no. 4, pp. 427–437, Jul. 2009, doi: 10.1016/j.ipm.2009.03.002.

[24] T. Fawcett, "Data Science for Business," 2013. [Online]. Available: https://www.researchgate.net/publication/256438799

[25] Brownlee Jason, “What is a Confusion Matrix in Machine Learning,” Machine learning mastery.