

# Prediction of Malware Threats using Machine Learning Techniques

RAED ALAZAIDAH<sup>1</sup>, GHASSAN SAMARA<sup>1</sup>, MOHAMMAD ALJAIDI<sup>1</sup>, MAIS HAJ QASEM<sup>1</sup>,  
ABDULLAH AL-QAMMAZ<sup>1</sup>, MOHAMMAD RASMI AL-MOUSA<sup>1</sup>, WAEL HADI<sup>2</sup>,

<sup>1</sup>Faculty of Information Technology,  
Zarqa University,  
JORDAN

<sup>2</sup>Faculty of Information Technology,  
University of Petra,  
JORDAN

*Abstract:* - Machine learning has been used for decades to analyze vast datasets, classify and cluster data, and make predictions using algorithms. One of its top use areas is cybersecurity, where it can help detect and prevent destructive threats such as malware. The use of machine learning in cybersecurity has proven to be a powerful tool in detecting and predicting malware attacks. In recent years, the number of Internet users has greatly increased and with it the number of malware attacks. This has made predicting malware a challenge. Consequently, to date, there is still a need to examine the numerous existing MLs' performance. This study is presented to identify the best classification model for predicting malware using two datasets and 18 different classifiers belonging to six learning strategies. The results showed that the RandomForest classifier had the highest accuracy, precision, recall, F1-measure, and ROC Area metrics, Moreover, Trees and Bayes learning strategies showed the best predictive performance on the two datasets compared with the other five learning strategies.

*Key-Words:* - Classification, Machine Learning, Malware, Prediction, Cyber Security, Malware Threats.

Received: March 19, 2025. Revised: June 12, 2025. Accepted: July 13, 2025. Available online: October 7, 2025.

## 1 Introduction

Malicious software (Malware) is a specifically designed software to damage, destroy, or even gain control over a computer's systems, networks, or servers, [1]. The term "malware" encompasses a wide range of applications, but most share common characteristics such as a payload that performs malicious actions. Malware is a major threat to cybersecurity and has become a significant concern for businesses, individuals, and even governments, [2]. According to AV-TEST (an independent German research institute for IT security), recent statistics showed that more than 500,000 malware pieces are detected daily, and more than 1 billion malware programs exist out there. Moreover, around 4 companies suffer from ransomware attacks every minute, [3], [4]. Therefore, the detection of malware is crucial to ensure the security of systems and the protection of sensitive information. Nevertheless, since the first malware was released in 1988, the number of new samples has been growing and

traditional methods of detecting malware such as signature scanning have become ineffective. To combat this problem, newer solutions based on the structure and behavior of malware have been proposed, such as using machine learning. Machine Learning (ML) algorithms can be used to extract important behavior characteristics and train a classifier, [5].

Be as it may, ML is a significant field of Artificial Intelligence (AI) that imitates intelligent human behavior, [5]. It has the capability to perform hard and complicated tasks such as classification, regression, clustering, and association analysis among several tasks, [6]. The main task of ML that is considered in this paper is classification. Classification is the accurate prediction of the class label for unseen cases, [7], [8], [9]. It is divided according to the number of classes associated with each case into either Single Label Classification (SLC) or Multi-label Classification (MLC), [7], [10], [11]. The former always associates only one class

label to any case or instance, while the latter may associate more than one class label to an instance or case. SLC itself is divided into subtypes, binary classification, and multi-class classification. Binary classification consists of two class labels only, while multi-class classification consists of more than two class labels, [12]. Therefore, this paper is interested in SLC.

The main objectives of this paper are to identify the most appropriate SLC algorithm for predicting malware among many algorithms that belong to different learning strategies and to identify the best learning strategy for predicting malware among six different well-known strategies.

The rest of this paper is organized as follows; Section 2 surveys related works to malware prediction. Section 3 introduces the methodology. Section 4 and Section 5 provide the results and discussion, respectively. Finally, Section 6 concludes the study and suggests future directions.

## 2 Related Work

Machine learning is a field of artificial intelligence that is closely associated with computational statistics, data mining, and data science, [13], [14]. It aims to teach computers to learn from data, using mathematical theories, statistical analysis, optimization, and various real-world applications. In cybersecurity, machine learning employs a data-driven approach, using raw security data to create an intelligent security model that can predict future incidents, [15], [16].

Authors in [17] reviewed the challenges and solutions related to using machine learning techniques for network intrusion detection systems. They observed that every technique has its advantages and disadvantages, and none of them can be considered the best without limitations. Data collection is a significant challenge due to its time-consuming and arduous nature, with publicly available datasets often being outdated or containing missing or redundant values. By contrast, the current paper addresses a wider range of cybersecurity threats and assesses machine learning models in those areas. In [18], authors classified malware based on static, dynamic, and hybrid analysis and reviewed various papers that used machine learning techniques to detect malware. However, the focus was solely on malware, and there was no critical analysis or performance evaluation of machine learning

techniques. Additionally, there was no explanation of the current state-of-the-art malware datasets. In comparison, our paper targets multiple cybersecurity threats, explains commonly used datasets, and presents performance evaluations of significant machine learning techniques on frequently used datasets.

Authors in [19] conducted a review of papers that used machine learning techniques for detecting cyber threats, but their focus was mainly on intrusion detection, and they didn't assess the performance of these techniques or provide any benchmark datasets. Authors in [20] surveyed the application of machine learning in cybersecurity, highlighting the challenges in using these techniques to combat cyber-attacks and threats. However, machine learning classifiers are prone to various cyber and adversarial attacks, and there is a significant need for improving their safety. Authors [21] examined publications on machine learning techniques in cyber security between 2008 and early 2016, finding that although the role of these techniques is increasing, it remains difficult to select the appropriate approach for specific safety issues.

In [22] authors assessed machine learning techniques for anomaly detection and feature selection in ML and observed that CNN classifiers could be more effectively applied to cybersecurity, but limitations such as missing or incorrect signatures impede detection. They recommended exploring knowledge-based and behavioral-based approaches and suggested further research in these areas. In another study [23], authors analyzed the use of machine learning for detecting spam, malware, and intrusions. They found that these techniques are vulnerable to cyber threats and face significant challenges. They emphasized the importance of finding a suitable classifier for specific safety issues and addressing the limitations of machine learning due to the increasing sophistication of cyber attackers.

In [24], researchers used a probability-based Bayesian network to classify events that process TCP/IP packets. Building on this, [25] designed a denial-of-service (DoS) intrusion detector using the same Bayesian network. The authors in [26] analyzed the KDD'99 cup dataset, which includes four attack categories—Probe or Scan, DoS, U2R, and R2L—using a probability-based naïve Bayes classifier.

In [27], researchers used the Naive Bayes classifier to construct a multi-class intrusion detection system. Other studies have employed the K-Nearest

Neighbors (KNN) algorithm, an instance-based learning algorithm that classifies a data point based on its proximity to the k-nearest neighbors. Authors in [28], [29] and [30] also implemented the KNN classification technique in their research to develop intrusion detection systems. Logistic regression has been used in various studies [31], [32] to detect malicious traffic and intrusions. Additionally, the authors of [33] explored a neural classifier, while those in [34] investigated the wavelet transform for detecting anomalies, particularly in the case of DoS attacks.

Decision trees are considered one of the most popular machine-learning techniques for building predictive models. The ID3 [35] and C4.5 [36] algorithms are commonly used to automatically construct decision trees, and in [37] recently proposed BehavDT, a behavioral decision tree algorithm for analyzing behavioral patterns. Many studies on cybersecurity have used decision tree classification to develop intrusion detection systems, including those in [4], [38], [39], [40] and [41]. However, decision tree models may encounter issues when dealing with high-dimensional security features, such as overfitting, high computational cost, and low prediction accuracy.

### 3 Methodology

The methodology of this research consists of four main steps. Figure 1 depicts the methodology followed in this research, providing a clear and concise overview of the entire process.

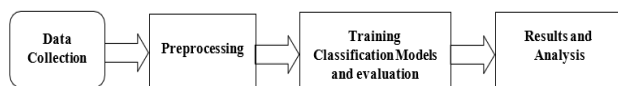


Fig. 1: Research Methodology

The first step is the data collection step, where two datasets with different characteristics have been chosen. We chose the datasets because they are highly relevant to our research topic and readily accessible, Section 3.1 provides a description of the datasets. In the second step, we clean and prepare the data for analysis. This involves removing instances with missing data and performing other necessary cleaning tasks. Table 1, included in the research, shows the characteristics of the datasets after preprocessing. The third step involves evaluating several classifiers from different learning strategies.

We use five well-known metrics—accuracy, precision, recall, F1-score, and AUC-ROC—to determine the best classifier for predicting malware.

In the fourth step, we analyze and interpret the results of the classifier evaluation. This may include comparing the performance of different classifiers, identifying patterns or trends in the data, and drawing conclusions about our research question. Finally, we wrap up the research by summarizing and discussing the findings, identifying limitations, and suggesting directions for future research.

#### 3.1 Datasets

This paper uses two datasets. The first one, called CLaMP (Classification of Malware with Portable headers), has 5,184 samples and 69 features. Each sample is labeled as either Malware or Benign. You can download the CLaMP dataset from Kaggle. It's great for training and testing different machine-learning models because it includes a wide range of malware features. The second dataset, named Malware, has 7,107 samples and 280 features. Like CLaMP, it also has two labels: malicious and non-malicious. This dataset can be found at the UCI Machine Learning Repository. It provides a lot of malware samples, making it ideal for realistic training and evaluation of machine learning models.

Table 1 in the paper provides a summary of the main characteristics of these datasets, such as the number of samples, features, and class labels. It also shows the data distribution and class imbalance, which are important for evaluating how well the machine learning models perform. Overall, the datasets considered in this research provide a valuable resource for researchers and practitioners working on malware classification and detection tasks.

Table 1. Dataset Characteristics

Name	Instances	Features	No. of Classes	Missing Values
CLaMP	5212	69	2	No
Malware	373	531	2	No

#### 3.2 Results

Depending on the undertaken experiments, the evaluation results are presented. To identify the best classifier that can accurately and instantly predict Malware, 18 different classifiers have been considered and evaluated using 5 different evaluation metrics. These classifiers belong to six learning

strategies. From Bayes's learning strategy, BayesNet, NaiveBayes, and NaiveBayes have been considered. Three different classifiers (Logistic, SMO, and SimpleLogistic) have been chosen to represent the Function strategy. Also, the Lazy learning strategy has been represented by three classifiers (IBK, KStar, and LWL). For Meta-learning strategy, AdaboostM1, LogitBoost, and MultiClassClassifier have been selected. Rules learning strategy has been represented through DecisionTable, JRip, and PART classifiers. Finally, the Tree learning strategy has been represented by RandomTree, RandomForest, and J48 classifiers.

Regarding the evaluation metrics [42], [43], 5 metrics have been considered (Accuracy, Precision, Recall, F1 measure, and ROC Area). These metrics are computed using the following equations:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+F} \quad (1)$$

$$precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$F1 - measure = 2 \cdot \frac{Precision}{recall} \quad (4)$$

The ROC (Receiver Operating Characteristics) metric is a graph that evaluates the performance of the classifier using all thresholds. ROC plots the FP (False Positive) rate as the X-axis, and plots the TP (True Positive) rate as Y-axis. TP rate and FP rate are calculated using the following two equations:

$$TP\ rate = \frac{TP}{TP + FN} \quad (5)$$

$$FP\ rate = \frac{FP}{FP + TN} \quad (6)$$

Table 2 (Appendix) depicts the evaluation results of the considered datasets using Accuracy, Precision, and Recall metrics.

According to Table 2 (Appendix), RandomForest achieved the best results on the CLaMP dataset considering Accuracy, Precision, and Recall metrics. For the Malware dataset, and considering the same metrics, BayesNet, NaiveBayes, and

NaiveBayesUpdateable achieved the best results with identical performance.

Moreover, considering the learning strategy, Trees showed the best results on the CLaMP dataset, while Bayes as a learning strategy showed the best results on the Malware dataset on the three considered evaluation metrics.

Table 3 (Appendix) depicts the evaluation results for F-measure and ROC Area metrics on CLaMP and Malware datasets.

Based on Table 3 (Appendix), RandomForest showed the best performance considering F-measure and ROC Area metrics on the CLaMP dataset and the best ROC Area result on the Malware dataset. For the Malware dataset, all Bayes-based classifiers showed an identical best performance considering the Accuracy metric.

Considering the learning strategy, Trees showed the best Accuracy average on the CLaMP dataset, while the Meta learning strategy showed the best ROC Area average on the same dataset. Bayes as a learning strategy showed the best averages for Accuracy and ROC Area metrics on the Malware dataset.

Figure 2 (Appendix) depicts the running time (per second) for the 18 classifiers considered in this dataset for both CLaMP and Malware datasets.

According to Figure 2 (Appendix), and considering the CLaMP dataset, the best running time has been achieved by the three lazy-based classifiers (IBK, KStar, LWL). SimpleLogistic and SMO classifiers from the Function Learning strategy showed the worst running time among the 18 considered classifiers on the CLaMP dataset.

For the Malware dataset, and based on Figure 2 (Appendix), also, the Lazy-based classifiers showed the best running time. DecisionTable from the Rules learning strategy and SimpleLogistic from the Function learning strategy showed the worst running time on the Malware dataset among the considered classifiers.

Among the four classifiers that showed the best performance using Accuracy, precision, Recall, F-measure, and ROC Area metrics, NaiveBayes showed the best running time on the CLaMP dataset and NaiveBayesUpdateable showed the best running time one Malware dataset. RandomForest showed a poor running time on both datasets.

Figure 3 (Appendix) depicts the running time (per second) with respect to the six learning strategies considered in this paper.

According to Figure 3 (Appendix), the best running time has been achieved by the Lazy learning strategy on both datasets. The Function learning strategy showed the worst running time on the CLaMP dataset, while the Rules learning strategy showed the worst running time on the Malware dataset.

Bayes showed better running time than Trees on the CLaMP dataset, while the case was the opposite on the Malware dataset.

#### 4 Discussion

According to the previous results in Appendix in Table 2 and Table 3, it can be clearly observed that the predictive performance of the 18 classifiers on the Malware dataset is much better than their predictive performance on the CLaMP dataset. The main reason for that is the nature of the datasets. All features in the Malware dataset are binary features, while in the CLaMP dataset the percentage of binary features is less than 45%. Accordingly, it can be concluded the significance of utilizing discretization and binarization techniques in datasets where a high percentage of its features are of numerical type.

Moreover, Based on Table 4 which depicts the Average for Accuracy, Precision, and Recall with respect to the learning strategy being used, it can be concluded that Trees showed the best performance on the CLaMP dataset, while Bayes showed the best performance on Malware dataset. Trees as a learning strategy was the third best learning strategy on the Malware dataset with a very slight difference from Bayes and Function learning strategies.

Nevertheless, Trees learning strategy was the least learning strategy affected by the type and range of the features in the dataset as can be observed from Table 4. Therefore, it can be concluded that Trees learning strategy is the best choice to handle any dataset regardless of its feature types.

Table 4. The average for Accuracy, Precision, and Recall concerning learning strategy

Learning Strategy	CLaMP			Malware		
	Accuracy	Precision	Recall	Accuracy	Precision	Recall
Bayes	74.376	0.830	0.743	<b>99.464</b>	<b>0.995</b>	<b>0.995</b>
Function	95.707	0.957	0.957	99.106	0.991	0.991
Lazy	91.734	0.922	0.917	98.570	0.986	0.986
Meta	94.946	0.950	0.950	98.928	0.989	0.989
Rules	96.929	0.969	0.969	98.928	0.989	0.989
Trees	<b>97.985</b>	<b>0.980</b>	<b>0.980</b>	99.017	0.990	0.990

#### 5 Conclusion and Future Work

In this paper, a comparative analysis among eighteen different classifiers that belong to six learning strategies has been conducted. Two datasets related to malware with different characteristics have been utilized in the analysis. Moreover, five evaluation metrics have been considered in addition to the time needed to build the classifier metric. The results showed the superior performance of the RandomForest classifier in comparison with the other classifiers, and the superiority of Trees as a learning strategy against the other five learning strategies. In a future work, more evaluation is suggested using other datasets and metrics.

#### Declaration of Generative AI and AI-assisted Technologies in the Writing Process

The authors wrote, reviewed and edited the content as needed and they have not utilised artificial intelligence (AI) tools. The authors take full responsibility for the content of the publication.

#### References:

- [1] De Gaspari, F., Hitaj, D., Pagnotta, G., De Carli, L., & Mancini, L. V. Evading behavioral classifiers: a comprehensive analysis on evading ransomware detection techniques. *Neural Comput & Applic* 34, 12077–12096 (2022). <https://doi.org/10.1007/s00521-022-07096-6>.
- [2] Alazaidah, R., Samara, G., Almatarneh, S., Hassan, M., Aljaidi, M., & Mansur, H. (2023). Multi-Label Classification Based on Associations. *Applied Sciences*, 13(8), 5081. <https://doi.org/10.3390/app13085081>.
- [3] Samara, G. (2020, November). Wireless sensor network MAC energy-efficiency protocols: a survey. In *2020 21st International Arab Conference on Information Technology (ACIT)* (pp. 1-5), Egypt. IEEE. <https://doi.org/10.1109/ACIT50332.2020.9300065>.
- [4] Al-Batah, Mohammad Subhi, Mazen Alzyoud, Raed Alazaidah, Malek Toubat, Haneen Alzoubi, and Areej Olaiyat. "Early Prediction of Cervical Cancer Using Machine Learning Techniques." *Jordanian Journal of Computers and Information Technology (JJCIT)* 8, no. 04, 357-369, (2022).

- [5] Samara, G. (2020). Intelligent reputation system for safety messages in VANET. *Int J Artif Intell*, 9(3), 439-447. doi: [doi.org/10.11591/ijai.v9.i3.pp439-447](https://doi.org/10.11591/ijai.v9.i3.pp439-447).
- [6] Alazaidah, R., Ahmad, F. K., & Mohsin, M. (2020). Multi label ranking based on positive pairwise correlations among labels. *The International Arab Journal of Information Technology*, 17(4), 440-449. doi: 10.34028/iajit/17/4/2.
- [7] Alazaidah, R., Ahmad, F. K., Mohsen, M. F. M., & Junoh, A. K. (2018). Evaluating conditional and unconditional correlations capturing strategies in multi label classification. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 10(2-4), 47-51.
- [8] Hussain, I., Samara, G., Ullah, I., & Khan, N. (2021, December). Oman. Encryption for end-user privacy: a cyber-secure smart energy management system. In *2021 22nd International Arab Conference on Information Technology (ACIT)* (pp. 1-6). IEEE. <https://doi.org/10.1109/ACIT53391.2021.9677341>.
- [9] Injadat, M. (2023, December). UAE. Optimized Ensemble Model Towards Secured Industrial IoT Devices. In *2023 24th International Arab Conference on Information Technology (ACIT)* (pp. 1-5), Ajman, United Arab Emirates. IEEE. <https://doi.org/10.1109/ACIT58888.2023.10453914>.
- [10] Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques*. Waltham: Morgan Kaufmann Publishers.
- [11] Injadat, M., Moubayed, A., Nassif, A. B., & Shami, A. (2020). Multi-stage optimized machine learning framework for network intrusion detection. *IEEE Transactions on Network and Service Management*, 18(2), 1803-1816. <https://doi.org/10.1109/TNSM.2020.3014929>.
- [12] Witten, I.H. and Frank, E., 2002. Data mining: practical machine learning tools and techniques with Java implementations. *Acm Sigmod Record*, 31(1), pp.76-77. <https://doi.org/10.1145/507338.507355>.
- [13] . Agrawal R, Gehrke J, Gunopulos D, Raghavan P. Fast algorithms for mining association rules. In: Proceedings of the International Joint Conference on Very Large Data Bases, Santiago Chile. 1994; 1215: 487-499.
- [14] Sarker, I.H. Context-aware rule learning from smartphone data: survey, challenges and future directions. *J Big Data* 6, 95 (2019), pp. 1-25. <https://doi.org/10.1186/s40537-019-0258-4>.
- [15] Salloum, S.A., Alshurideh, M., Elnagar, A. and Shaalan, K., 2020. Machine learning and deep learning techniques for cybersecurity: a review. In *Proceedings of the International Conference on Artificial Intelligence and Computer Vision (AICV2020)* (pp. 50-57), Cairo, Egypt. Springer International Publishing. [https://doi.org/10.1007/978-3-030-44289-7\\_5](https://doi.org/10.1007/978-3-030-44289-7_5).
- [16] Gandotra, E. and Gupta, D., 2021. An efficient approach for phishing detection using machine learning. *Multimedia Security: Algorithm Development, Analysis and Applications*, pp.239-253. Springer. [https://doi.org/10.1007/978-981-15-8711-5\\_12](https://doi.org/10.1007/978-981-15-8711-5_12).
- [17] Dharamkar, B. and Singh, R.R., 2014. A review of cyber attack classification technique based on data mining and neural network approach. *Int. J. Comput. Trends Technol*, 7(2), pp.100-105. doi: 10.14445/22312803/IJCTT-V7P106.
- [18] Ford, V. and Siraj, A., 2014, October. Applications of machine learning in cyber security. In *Proceedings of the 27th international conference on computer applications in industry and engineering* (Vol. 118). Kota Kinabalu, Malaysia: IEEE Xplore.
- [19] Jiang, H., Nagra, J. and Ahammad, P., 2016. SoK: Applying Machine Learning in Security-A Survey. *arXiv e-prints*, pp.arXiv-1611. <https://doi.org/10.48550/arXiv.1611.03186>.
- [20] Hodo, E., Bellekens, X., Hamilton, A., Tachtatzis, C. and Atkinson, R., 2017. Shallow and deep networks intrusion detection system: A taxonomy and survey. *arXiv preprint arXiv:1701.02145*.
- [21] Apruzzese, G., Colajanni, M., Ferretti, L., Guido, A. and Marchetti, M., 2018, May. On the effectiveness of machine and deep learning for cyber security. In *2018 10th international conference on cyber Conflict (CyCon), Tallinn, Estonia.* (pp. 371-390). IEEE. <https://doi.org/10.23919/CYCON.2018.8405026>.
- [22] Kruegel, C., Mutz, D., Robertson, W. and

- Valeur, F., 2003, December. USA. Bayesian event classification for intrusion detection. In *19th Annual Computer Security Applications Conference, 2003. Proceedings. Las Vegas, NV, USA.* (pp. 14- 23). IEEE. <https://doi.org/10.1109/CSAC.2003.1254306>.
- [23] Benferhat, S., Kenaza, T. and Mokhtari, A., 2008, July. Finland. A naive bayes approach for detecting coordinated attacks. In *2008 32nd Annual IEEE International Computer Software and Applications Conference* (pp. 704- 709), Turku, Finland. IEEE. <https://doi.org/10.1109/COMPSAC.2008.213>.
- [24] Panda, M. and Patra, M.R., 2007. Network intrusion detection using naive bayes. *International journal of computer science and network security*, 7(12), pp.258-263.
- [25] Koc, L., Mazzuchi, T.A. and Sarkani, S., 2012. A network intrusion detection system based on a Hidden Naïve Bayes multiclass classifier. *Expert Systems with Applications*, 39(18), pp.13492-13500. <https://doi.org/10.1016/j.eswa.2012.07.009>.
- [26] Shapoorifard, H. and Shamsinejad, P., 2017. Intrusion detection using a novel hybrid method incorporating an improved KNN. *Int. J. Comput. Appl*, 173(1), pp.5-9. doi: 10.5120/ijca2017914340.
- [27] Vishwakarma, S., Sharma, V. and Tiwari, A., 2017. An intrusion detection system using KNN-ACO algorithm. *Int J Comput Appl*, 171(10), pp.18-23. doi: 10.5120/ijca2017914079.
- [28] Sharifi, A.M., Amirgholipour, S.K. and Pourebrahimi, A., 2015. Intrusion detection based on joint of k-means and knn. *Journal of Convergence Information Technology*, 10(5), p.42.
- [29] Bapat, R., Mandya, A., Liu, X., Abraham, B., Brown, D.E., Kang, H. and Veeraraghavan, M., 2018, April. USA. Identifying malicious botnet traffic using logistic regression. In *2018 systems and information engineering design symposium (SIEDS), Charlottesville, VA, USA.* (pp. 266-271). IEEE. <https://doi.org/10.1109/SIEDS.2018.8374749>.
- [30] Besharati, E., Naderan, M. and Namjoo, E., 2019. LR-HIDS: logistic regression host-based intrusion detection system for cloud environments. *Journal of Ambient Intelligence and Humanized Computing*, 10, pp.3669-3692. <https://doi.org/10.1007/s12652-018-1093-8>.
- [31] Kumar, P.A.R. and Selvakumar, S., 2011. Distributed denial of service attack detection using an ensemble of neural classifier. *Computer Communications*, 34(11), pp.1328-1341. <https://doi.org/10.1016/j.comcom.2011.01.012>.
- [32] Dainotti, A., Pescapé, A. and Ventre, G., 2009. A cascade architecture for DoS attacks detection based on the wavelet transform. *Journal of Computer Security*, 17(6), pp.945-968. <https://dl.acm.org/doi/abs/10.5555/1662641.1662646>
- [33] Quinlan, J.R., 1986. Induction of decision trees. *Machine learning*, 1, pp.81-106. <https://doi.org/10.1007/BF00116251>.
- [34] Quinlan, J.R., 2014. *C4. 5: programs for machine learning*. Elsevier. <https://dl.acm.org/doi/abs/10.5555/152181>.
- [35] Sarker, I.H., Colman, A., Han, J., Khan, A.I., Abushark, Y.B. and Salah, K., 2020. Behavdt: a behavioral decision tree learning to build user-centric context-aware predictive model. *Mobile Networks and Applications*, 25, pp.1151-1161. <https://doi.org/10.1007/s11036-019-01443-z>.
- [36] Ingre, B., Yadav, A. and Soni, A.K., 2018. Decision tree based intrusion detection system for NSL-KDD dataset. In *Information and Communication Technology for Intelligent Systems (ICTIS 2017)-Volume 2 2* (pp. 207-218). Springer International Publishing. [https://doi.org/10.1007/978-3-319-63645-0\\_23](https://doi.org/10.1007/978-3-319-63645-0_23).
- [37] Malik, A.J. and Khan, F.A., 2018. A hybrid technique using binary particle swarm optimization and decision tree pruning for network intrusion detection. *Cluster Computing*, 21, pp.667-680. <https://doi.org/10.1007/s10586-017-0971-8>.
- [38] Moon, D., Im, H., Kim, I. and Park, J.H., 2017. DTB-IDS: an intrusion detection system based on decision tree using behavior analysis for preventing APT attacks. *The Journal of supercomputing*, 73, pp.2881-2895. <https://doi.org/10.1007/s11227-015-1604-8>.
- [39] Puthran, S. and Shah, K., 2016. Intrusion detection using improved decision tree algorithm with binary and quad split. In *Security in Computing and Communications: 4th International Symposium, SSCC 2016*,

Jaipur, India, September 21-24, 2016, *Proceedings 4* (pp. 427-438). Springer Singapore. [https://doi.org/10.1007/978-981-10-2738-3\\_37](https://doi.org/10.1007/978-981-10-2738-3_37).

- [40] Alhusenat, A. Y., Owida, H. A., Rababah, H. A., Al-Nabulsi, J. I., & Abuowaida, S. (2023). A Secured Multi-Stages Authentication Protocol for IoT Devices. *Mathematical Modelling of Engineering Problems*, 10(4). pp. 1352-1358.
- [41] Owida, H. A., Migdadi, H. S., Hemied, O. S. M., Alshdaifat, N. F. F., Abuowaida, S. F. A., & Alkhawaldeh, R. S. (2022). Deep learning algorithms to improve COVID-19 classification based on CT images. *Bulletin of Electrical Engineering and Informatics*, 11(5), 2876-2885. <https://doi.org/10.11591/eei.v11i5.3802>.
- [42] Owida, H. A., Moh'd, B. A. H., Turab, N., Al-Nabulsi, J., & Abuowaida, S. (2023). The Evolution and Reliability of Machine Learning Techniques for Oncology. *International Journal of Online & Biomedical Engineering*, 19(8), pp. 110–129.. doi: [doi.org/10.3991/ijoe.v19i08.39433](https://doi.org/10.3991/ijoe.v19i08.39433).
- [43] Owida, H. A., Hemied, O. S. M., Alkhawaldeh, R. S., Alshdaifat, N. F. F., & Abuowaida, S. F. A. (2022). Improved deep learning approaches for covid-19 recognition in ct images. *Journal of Theoretical and Applied Information Technology*, 100(13), 4925-4931.

### **Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)**

The authors equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

### **Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself**

This research is funded by the Deanship of Research and Graduate Studies in Zarqa University /Jordan

### **Conflict of Interest**

The authors have no conflicts of interest to declare.

### **Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)**

This article is published under the terms of the Creative Commons Attribution License 4.0

[https://creativecommons.org/licenses/by/4.0/deed.en\\_US](https://creativecommons.org/licenses/by/4.0/deed.en_US)

## APPENDIX

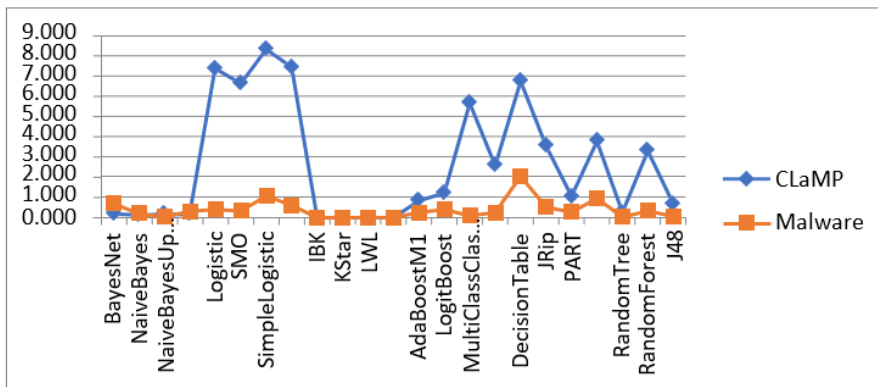


Fig. 2: Running time for the considered 18 classifiers

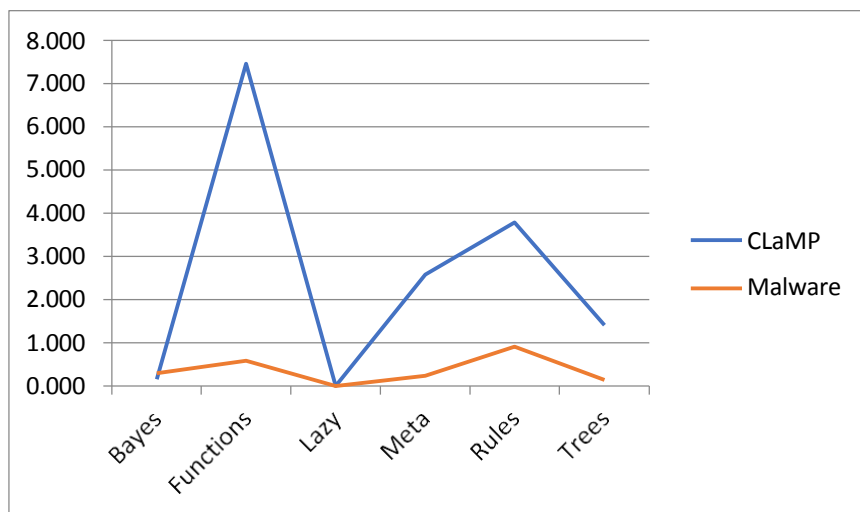


Fig.3: Running time for the considered learning strategies

Table 2. Evaluation Results Using Accuracy, Precision, and Recall

Learning Strategy	Classifier	CLaMP			Malware		
		Accuracy	Precision	Recall	Accuracy	Precision	Recall
Bayes	BayesNet	94.837	0.948	0.948	<b>99.464</b>	<b>0.995</b>	<b>0.995</b>
	NaiveBayes	64.146	0.771	0.641	<b>99.464</b>	<b>0.995</b>	<b>0.995</b>
	NaiveBayesUpdateable	64.146	0.771	0.641	<b>99.464</b>	<b>0.995</b>	<b>0.995</b>
	<b>Average</b>	74.376	0.830	0.743	<b>99.464</b>	<b>0.995</b>	<b>0.995</b>
Functions	Logistic	96.392	0.964	0.964	99.196	0.992	0.992
	SMO	94.472	0.945	0.945	99.196	0.992	0.992
	SimpleLogistic	96.257	0.963	0.963	98.928	0.989	0.989
	<b>Average</b>	95.707	0.957	0.957	99.106	0.991	0.991
Lazy	IBK	97.140	0.971	0.971	98.660	0.987	0.987
	KStar	86.948	0.882	0.869	98.660	0.987	0.987
	LWL	91.113	0.913	0.911	98.391	0.984	0.984
	<b>Average</b>	91.734	0.922	0.917	98.570	0.986	0.986
Meta	AdaBoostM1	93.685	0.937	0.937	98.660	0.987	0.987
	LogitBoost	94.760	0.948	0.948	98.928	0.989	0.989
	MultiClassClassifier	96.392	0.964	0.964	99.196	0.992	0.992
	<b>Average</b>	94.946	0.950	0.950	98.928	0.989	0.989
Rules	DecisionTable	94.971	0.950	0.950	98.928	0.989	0.989
	JRip	97.716	0.977	0.977	98.928	0.989	0.989
	PART	98.100	0.981	0.981	98.928	0.989	0.989
	<b>Average</b>	96.929	0.969	0.969	98.928	0.989	0.989
Trees	RandomTree	96.833	0.968	0.968	98.928	0.989	0.989
	RandomForest	<b>99.098</b>	<b>0.991</b>	<b>0.991</b>	99.196	0.992	0.992
	J48	98.023	0.980	0.980	98.928	0.989	0.989
	<b>Average</b>	<b>97.985</b>	<b>0.980</b>	<b>0.980</b>	99.017	0.990	0.990

Table 3. Evaluation Results Using F-measure and ROC Area Metrics

Learning Strategy	Classifier	CLaMP		Malware	
		F-measure	ROC Area	F-measure	ROC Area
Bayes	BayesNet	0.948	0.985	<b>0.995</b>	0.993
	NaiveBayes	0.602	0.955	<b>0.995</b>	0.993
	NaiveBayesUpdateable	0.602	0.955	<b>0.995</b>	0.993
<b>Average</b>		0.717	0.965	<b>0.995</b>	<b>0.993</b>
Functions	Logistic	0.964	0.989	0.992	0.979
	SMO	0.945	0.944	0.992	0.984
	SimpleLogistic	0.963	0.993	0.989	0.999
<b>Average</b>		0.957	0.975	0.991	0.987
Lazy	IBK	0.971	0.971	0.986	0.971
	KStar	0.869	0.925	0.986	0.984
	LWL	0.911	0.945	0.984	0.999
<b>Average</b>		0.917	0.947	0.985	0.985
Meta	AdaBoostM1	0.937	0.978	0.986	0.999
	LogitBoost	0.948	0.989	0.989	0.999
	MultiClassClassifier	0.964	0.989	0.992	0.979
<b>Average</b>		0.950	<b>0.985</b>	0.989	0.992
Rules	DecisionTable	0.950	0.976	0.989	0.964
	JRip	0.977	0.982	0.989	0.971
	PART	0.981	0.987	0.989	0.974
<b>Average</b>		0.969	0.981	0.989	0.970
Trees	RandomTree	0.968	0.969	0.989	0.978
	RandomForest	<b>0.991</b>	<b>0.999</b>	0.992	<b>0.999</b>
	J48	0.980	0.986	0.989	0.973
<b>Average</b>		<b>0.980</b>	0.984	0.990	0.983