Systematic Literature Review on Malware Detection and Machine Learning Algorithms: Identifying Gaps for possible Remedies.

BARTHOLOMEW IDOKO[1], FRANSCISCA OGWUELEKA[2], STEVEN BASSEY[3]
[1,3]Centre for Cyber Space Studies, Nasarawa State University Keffi, NIGERIA.

[2]Computer Science Department, University of Abuja, Abuja-FCT, NIGERIA.

Abstract: The inability of the traditional malware detection systems to accurately detect and classify instances of malware attacks has become a problem that requires in-depth research. Consequently, Machine Learning (ML) based malware detection system could be a better tool to achieve the expected objectives. This study is a systematic review of malware analysis and detection in four (4) different citation databases and considers the total of 262 research articles published from 2014 to 2024. The methodology adopted in the study include evaluation and validation, search strategy, inclusion and exclusion criteria, selection procedure, and data extraction from the selected articles. The aim of the study is to analyze the papers published in the four citation databases based on Machine learning tasks (regression or classification), research methodology and ML algorithms used by the different authors. The results were presented as classification or regression functions and validated using bar charts and pie charts. The common objectives and anomaly in the detection scenarios were analyzed and gaps identified. The study will serve as a guide to researchers for decision making with regards to developing the best ML algorithm that could solve malware detection problems.

Key-words: Malware, detection; malware, analysis; Machine Learning; accuracy; model; classification.

Received: April 11, 2025. Revised: May 12, 2025. Accepted: June 2, 2025. Published: June 12, 2025.

1. Introduction

Machine Learning ML, and Deep Learning, DL are among the many subfields within the broad and rapidly expanding subject of artificial intelligence (Bush & Abiyev, 2023). In a nutshell, machine learning (ML) is the techniques that computers employed to learn from data to become more imaginative and predictive, precisely mimicking human brain (Helwan et al., 2017). ML research has advanced in nearly every branch of sciences, engineering, social sciences, and medicine. For the past 15 years, attackers have taken advantage of the inherent asymmetry in signature-based antivirus by altering few lines of code and making the compiled code look entirely different thereby breaking the system's defenses to compromise the system (Idoko et al., 2022; Idoko & Bush, 2023).

Consequently, the interest in the research in malware detection using ML has rose exponentially (Johnson & Grumbling, 2019). Several research are being conducted in malware analysis and detection using ML resulting in significant increase in papers being published in the field (Chen & Guestrin, 2016; Perrotta, & Selwyn, 2019).

This paper serves as a guide for researchers who intends to develop malware detection systems for real-life scenarios by examining the advancements and weaknesses in the current machine learning systems for malware detection from a wider perspective and suggesting solutions. As a result, an overview of the ML models, detection methods, and assessment standards frequently used in ML research has been provided.

Secondly, from 2014 to 2024, a comprehensive evaluation of the literature was carried out to determine the objectives of studying malware detection using ML, the frequency of usage of machine learning techniques by authors and detection methods. All these parameters among others were calculated as percentages and presented in the report.

The order of arrangement of the paper is such that: section 2 dealt with the fundamentals and theories of ML models in malware analysis and detection. section 3 presents the method and techniques used for the research. section 4 presents the results and analysis of the study and section 5 dealt with the conclusion and recommendation.

2. ML Techniques and Functions

This part of the paper gives the general overview of the popular machine learning techniques in malware detection and highlighted the basic ML functions.

2.1. Machine Learning Model

Three (3) major learning mechanisms are involved in machine learning. These are:

i. Supervised learning: Training robots and machines to learn under supervision using labeled data. Giving computers access to vast volumes of data and teaching them how to comprehend through the process of training (Sekeroglu et al., 2021; Uwanuakwa et al., 2022). For example, a variety of images of dogs from various perspectives are displayed to the computer, showcasing a variety of breeds, color variations, and other variations. If computers are to be able to analyze the information from these various dog photos, their intelligence must grow. The algorithm should eventually be able to determine whether a given image is of a dog from a completely different visual representation that wasn't available in the labeled data set of dog photos it was previously fed.

ii. Unsupervised learning: This learning platform on the other hand assess objects, images or substances that is not yet labeled. That is, training the system to understand as well as draw conclusions from dataset whose significance is invisible to human. The system looks for patterns or incidences (behaviour) in the dataset and uses those patterns to guide its own conclusions as shown in figure 1. It is important to note that computers using an unlabeled dataset produced the results that is invisible to human (Bush et al., 2018).

Figure 1: Behavioural Component of ML

iii. Reinforcement learning: This learning techniques relies on input. The method involves feeding a set of data to the machine and asking it to guess what it might be. If the system deduces the wrong conclusion from the incoming data, it gets feedback regarding its misclassification (Sekeroglu et al., 2021). This happens when a system automatically learns to detect an image of say a volleyball when it comes across a completely different image. For instance, if you give it an image of a volleyball and it incorrectly classify the volleyball as a hand ball or anything other than volleyball. Some of the ML models and their respective task are displayed in Table 1.

Table1. The commonly used machine learning model in malware detection and their functions.

S/N	Model	Function(s)
1	Naive Bayes Classifier (NBC)	Classification
2	Support Vector Machine	Classification
3	Multi-Class Support Vector Machine	Classification
4	Logistic Regression	Classification
5	Linear Regression	Regression
6	Support Vector Regression	Regression
7	Restricted Boltzmann Machine	Classification, Regression
8	Random Forest	Classification, Regression
9	Convolutional Neural Network	Classification, Regression
10	Artificial Neural Networks	Classification, Regression
11	Deep Neural Networks	Classification, Regression
12	Long Short-Term Memory Neural Networks	Classification, Regression
13	Extreme Gradient Boosting	Classification, Regression

Table 1 outlined some of the various machine learning models and their corresponding task. However, the detail study of some selected models that can be applied in malware detection cannot be overemphasized:

2.1.1. Naive Bayes classifier: Text categorization and other classification applications uses this well-known supervised machine learning technique. It is frequently seen as belonging to the family of generative learning algorithms, which suggests that it mimics the distribution of inputs for a certain class or category. Naive Bayes is a simple approach to classifier construction. These models represent algorithms with instances that uses feature vectors and classify them according to a limited number of class labels (Sekeroglu et al., 2019).

2.1.2. Support Vector Machine (SVM): This system functions as a classifier (Sekeroglu et al.,

2021). After the data has been categorized and mapped into hyperspace with the aid of a kernel function, the Support Vector Machine (SVM) assigns the vectors that are the closest data points of each class to one another (Caruana1 et al., 2011). To shorten the distance between the input, data points and the hyperplane, the classification procedure makes use of a subset generated by the input data's support vectors. One of the main goals of this method is to maximize the distance between the support vectors and the separating hyperplane; the ideal hyperplane is the one with the largest gap.

2.1.3. Multi-class Support Vector (MCSVM): Support Vector Machines (SVM) are a popular method for resolving binary classification problems in machine learning models (Kannadhasan et al., 2022). Multi-class SVMs (MCSVM) are usually created by combining multiple binary SVMs. As a matter of fact, MCSVM makes classification and regression easier. In this type of SVM, the computer assigns an instance to one of three classes or more.
Examples of multiclass classification include the followings:

a. Assigning a textual classification as positive, negative, or neutral
b. Identifying the type of malware included in a dataset of malware images
c. Sorting news articles into social, political, economic, or athletic categories.

2.1.4. Logistic Regression: The logistic regression method is only applicable to classification problems although it can liken to linear regression in many ways, its applicability is restricted to classification issues (Mason et al., 2018).

2.1.5. Random Forest (RF): This is a classification-based machine learning method. RF can construct multiple decision trees using a random subset selection of the data set (Otoum et al., 2020). It is conceptualized as a combination of tree predictors, in which each tree is independent of the distribution of all the trees in the generated forest and is dependent on the values of the random vectors. RF is usually used as the base classifier in hybrid models (Idoko & Nwankwo, 2025; Breiman, 2001). The following axioms are used in the creation of the RF algorithm:

a. Random trees r, are created to generate RF.
b. All of r's test feature results are merged.
c. The overall prediction is computed by considering the output from every tree.

Both regression and classification applications make substantial use of RF (Pahlavan-Rad et al., 2018; Yang et al., 2020).

2.1.6. Convolutional Neural Network (CNN): In this learning model, the input datasets are processed through multiple layers which is based on learning representation. This enables the framework to learn on its own and recognize the traits required for detection (Oytun et al., 2020). Deep learning has proved to outperform human reasoning in formal demonstrations on image and audio identification problems (Dougherty, 2013; Oytun, et al., 2020). A CNN is a multilayer neural network (NN) architecture that consists of one or more convolution, max-pooling, and fully connected layers. The convolution layers which are the basic building blocks of the network, are arranged in a hierarchical fashion.

2.1.7. Artificial Neural Networks (ANN): ANN frequently simulates the biological characteristics of the human brain and replicates human cognitive processes on computers. ANN comprises of the input, hidden, and output layers. The number of neurons in each layer varies depending on the application and parameter adjustments. In supervised learning, the estimated error propagates back to modify the weights and minimize the error in the ANN output (Sekeroglu & Dimililer, 2020).

2.1.8. Deep Neural Networks (DNN): This consists of multiple hidden layers positioned in between an ANN's input and output layers. Like shallow ANNs, DNNs can simulate complex non-linear interactions. The main purpose of a neural network is to handle real-world problems like categorization by taking in a set of inputs, processing them through more complex calculations, and then producing an output. An input, an output, and a sequential data flow are all present in a deep neural network (Bush et al., 2023).

2.1.9. Extreme Gradient Boosting: Another ensemble tree technique that boosts weak learners is called Extreme Gradient Boosting. It makes use of the gradient descent process, which is liken to the gradient boosting algorithm. Extreme Gradient Boosting (XGBoost) uses a variety of regularization models, including Least Absolute Shrinkage and Selection Operator (LASSO), to solve overfitting problems throughout the learning process (Chen et al., 2020; Ozsahin, et al., 2024; Gofwen et al., 2023). Additionally, each iteration determines the exact

number of iterations on a single run using built-in cross-validation.

3. Explored Techniques and Methodology

Evaluation and validation, search strategy, inclusion and exclusion criteria, selection procedure, and data extraction are some of the techniques/methods used in this work for a thorough systematic literature review.

3.1. Evaluation and Validation Techniques

The assessment measures for the regression and classification tasks differs because the models generate distinct outputs. Nonetheless, the methods used for validating both problem areas are comparable in terms data split (training and testing sets).

3.1.1. Classification Metrics

Malware detection classification studies, which seek to ascertain detection accuracy based on classes (e.g., True positive, Pass/Fail, etc.), typically take accuracy into account based on correctly and incorrectly categorized samples (Damodaran, et al., 2017). By dividing the total number of correctly classified samples by the total number of samples in the test set, the accuracy is calculated. However, if the dataset is unbalanced, accuracy is limited.

The accuracy result makes it impossible to evaluate the models if any class or dataset output has a disproportionately high or low number of samples compared to the other classes (Moustafa, et al., 2017). The accuracy formula can be expressed using equation (1).

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

Where the model's true positive, false positive, true negative, and false negative values are indicated by the letters TP, FP, TN, and FN respectively. The receiver operating characteristics area under curve (ROC AUC) is one of the most widely used metrics, particularly for two-class imbalanced data, despite the fact that there are still uncertainties in quantifying the outcomes produced on imbalanced data (Maigida, et al., 2019). The F1 score is another widely used statistic. It is described as the precision and recall harmonic means (Maigida, et al., 2019). One of the measures frequently applied to binary and multiclass problems with unbalanced data is the F1 score. Equation (2) shows the F1 score formula.

$$\text{F1 score} = \frac{TP}{TP + ½ (FP + FN)} \quad (2)$$

Recall (sensitivity), specificity, and accuracy are the other assessment metrics of classification tasks that are used to evaluate the models' unique capacities for identifying distinct output classes. The following equations specifically define some of the evaluation metrics considered in the study:

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

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (4)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (5)$$

3.1.2. Regression Metrics

Predicting the raw outcomes of anomaly occurrences in malware samples is the main goal of regression research in malware detection. The evaluation of the models is dependent on real-valued data, typically considering the difference between the predicted and observed data, because the samples in regression problems are not assigned to a particular class. The most widely used metrics are the Coefficient of Determination (R^2 score), Mean Squared Error (MSE), and Mean Absolute Error (MAE). By squaring the difference between the targeted and real data, MSE takes outliers into account more than other metrics. On the other hand, acceptable errors could result in an overestimate of the error (Venkatraman, et al., 2019). Equation (6) shows the MSE formula.

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \dot{y}_i)^2 \quad (6)$$

Where yi and ẏi is the actual and projected values, respectively, and N stands for the dataset's samples. The magnitude of the discrepancies between the observed and expected data is measured by MAE. Unlike MSE, where the direction of the error is not considered. With MAE, more reliable outcomes could be achieved (Oytun, et al., 2020). Equation (7) shows the MAE formula.

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} [y_i - \dot{y}_i] \quad (7)$$

The scaled correlation level between the observed and anticipated data is known as the R^2 score. In general, this enables researchers to acquire and

examine the evaluation results with greater rigor (Ever, et al., 2020). Equation (8) gives the R^2 score formula.

$$R^2 \text{ Score} = 1 - \frac{\sum (y_i - \dot{y}_i)}{\sum (y_i - \dot{y}_i)} \quad (8)$$

Where, $\dot{y}_i$ equals the mean of the dataset samples.

3.1.3 Validation Techniques

Machine learning models can be validated by applying a variety of methods. Despite the significant differences in these approaches, the hold-out, cross-validation and data mining techniques have been used to validate models throughout their intersection (Dougherty, 2013). Data mining techniques enable direct data selection for training by identifying the instances or qualities; even little changes to the training data can have a significant effect on the results. If training data are not considered using data selection techniques, then the cross-validation method is the most effective strategy in this situation (Zheng, et al., 2022). Cross-validation shows the full capability because all the data are utilized for both model training and testing, whereas the hold-out technique only uses dataset segmented in the training (70%) and testing (30%) phases independently (Zheng, et al., 2022). Cross-validation yields more accurate results on the models' abilities since it splits the dataset into k divisions and iteratively uses each partition during the training and testing stages (Zheng, et al., 2022). Cross-validation also has the benefit of being able to be used for hyper-parameter tuning. This offers a quicker way to adjust the model's parameters than the hold-out method.

3.2. Search Strategy

In order to illustrate the prevalence of validation, assessment metrics, and ML method used in the study investigations in percentages, a systematic literature review was conducted. As a result, this demonstrates how the outcomes could differ based on the datasets and techniques employed. To illustrate the growing interest in malware research and detection, the number of articles in the databases under consideration was obtained. The PRISMA statement was taken into consideration during the study.

Four citation databases; Google Scholar, IEEE Xplore, Web of Science, and Scopus were exploited for research published between January 1, 2014, and December 31, 2024. Three search phrases were used in the literature search: “Malware Analysis”, “Detection”, and “Machine Learning”. The search query implemented was (Malware Analysis AND Detection OR Machine Learning) to spool studies on malware analysis and detection using Machine Learning models.

3.3. Criteria for Inclusion and Exclusion

The articles used for the inclusion criteria met the following **requirements:**

i. Research articles (printed in peer-reviewed scientific journals).

ii. Research that used deep learning and/or machine learning methods for the purpose of malware analysis and detection in critical environments.

iii. The research written in English language.

The followings were established as the exclusion criteria:

i. Research that used methods other than machine learning for malware identification.

ii. Research papers on literature reviews, abstracts, book chapters, editorials, and commentary.

3.4. Data Extraction and Selection Process

The relevant research papers were chosen based on the inclusion and exclusion criteria. Figure 2 is a flow chart of the research selection procedure.

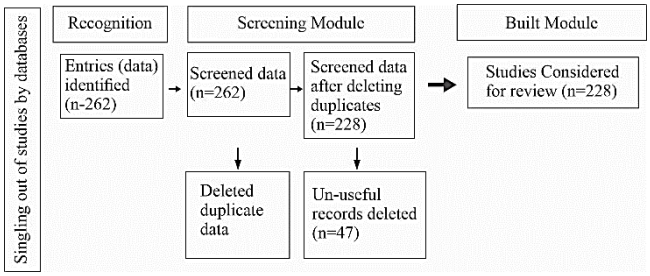


Figure 2: Selection procedure Flow chart

Data were extracted based on Three (3) objectives: (i) Machine learning tasks (regression or classification) (ii) Methodology or Techniques, (iii) Machine Learning Algorithms. Twenty (20) studies were extracted and included in this evaluation for presentation. The studies were chosen to address and cover all the study's goals from a wide perspective, including aim, various methods for evaluation and validation, and the ML models. The features of the studies chosen for presentation and further research are shown in Table 2.

4. Results and Analysis

The studies selected for investigation is presented and analyzed in this section.

4.1. Features of the Selected Studies

The features considered for the selected studies for further review include, Author/year of publication, Objective, ML Techniques, Evaluation Matric(s) and Validation Method as presented in table 2.

Table 2: Features of the Selected Studies

S/N	Author (s) & years of publication	Objective (s)	ML Techniques	Evaluation metric (s)	Validation method
1.	Johnson & Gumbling, 2019	Development of ML model for malware data	RF, DT	Recall, F1 precision, RMSE	Hold-out
2.	Zheng, et al, 2022	Malware classification	KNN, Logistic regression	Accuracy, F1 score, Recall	K- fold cross-validation
3.	Sign & Jain, 2017	Malware detection & classification	DNN, DT	Recall, accuracy	Hold-out
4.	Selmat et al, 2019	Comparative study of performance of ML models in malware detection	KNN, DT, SVM	Accuracy, ROC AUC	K-fold cross-validation
5.	Moustafa, et al., 2017	Malware classification	SVM, RF	Precision, RMSE	K-fold cross-validation
6.	Ambusaidi et al., 2016	Malware detection & classification	SVM	Accuracy, Precision, RMSE	-
7.	XU et al., 2017	ML framework for malware monitoring & classification	DT	F1 Score, Recall	Hold-out
8.	Liu, et al., 2017	Malware detection & Analysis	KNN	ROC AUC, Accuracy & recall	-
9.	Zhong & Gu., 2019	Malware detection	DNN, ANN	Recall, precision accuracy	K-fold cross-validation
10.	Mahindru & Sanga, 2020	Detecting malware on smart phones	DT, RF	Accuracy, ROC AUC	Hold-out
11.	Venkatraman, et al., 2019	Malware classification	CNN	Recall, precision, accuracy	Hold-out
12.	Rafique, et al., 2019	Malware detection & classification	CNN LSTM	Precision, F1 score	K-fold cross-validation
13.	Agarap, 2018	Malware Classification	L2-SVM	Accuracy	Hold-out
14.	Watson, et al., 2016	Detecting malware in cloud environment	ANN	Accuracy, ROC AUC, recall	K-fold cross-validation
15.	Moustafa & Hujillslay, 2019	Malware analysis	DT, CNN, SVM	Accuracy, precision	-
16.	Cen, et al., 2015	Malware detection	KNN, SVM	Accuracy	Hold-out
17.	Kim, et al., 2019	Android malware detection	CNN	Precision, accuracy	-
18.	Azeez, et al., 2021	Malware classification	DT	Recall	K-fold cross-validation
19.	Maigida et al., 2019	Detection & classification	ANN	Precision & recall	K-fold cross-validation
20.	Salehi, et al., 2021	Detection & classification	RF, SVM	Accuracy	K-fold cross-validation

4.2. Numbers and year of Selected Publications

After removing duplicate studies and papers that satisfied the exclusion criteria, the sum of all journal articles retrieved from the four citation databases between 2014 and 2024 dropped to 228. Figure 3 illustrates how research on machine learning-based malware analysis and detection began to garner interest after 2018. Between 2014 and 2016, only 14% (32/228) of the unduplicated journals were published, whereas 86% (196/228) were published between 2018 and 2024. The number of publications in the citation databases on yearly bases as well as the publication ratios for the specified years are presented in Figure 3.

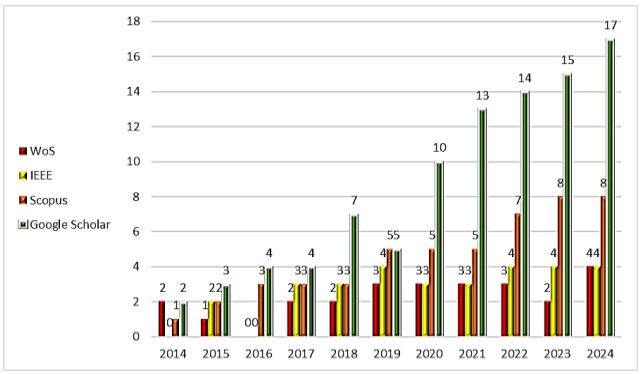


Fig. 3: Number of publications (2014 – 2024) n=228

4.3. Objective of the Selected Studies

The objectives of the studies in this context depicts the aim of the selected papers which in most cases were tailored towards malware analysis and/or malware detection. Figure 4 shows the computed percentages of classification and regression tasks as it applies to the studies (papers) that aim at malware analysis or malware detection or both.

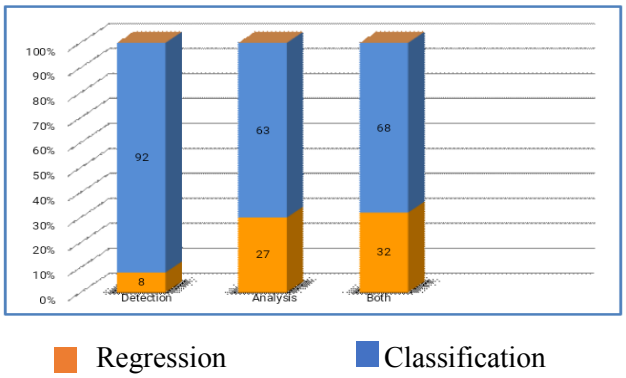


Figure 4: Core objectives of malware analysis and detection studies using ML

4.4. Frequency of Evaluation Matric (s)

Various evaluation measures allow researchers to gain insight into the models' performance in relation to the various factors they considered. The data in the flow chart in figure 2 can be used to determine the frequency of examined matrices for regression and classification. In regression research, RMSE and MAE were the most used evaluation measures which are responsible for 32% (14/47) and 26% (12/47) of the total. Remarkably, one of the least used metrics in the malware analysis and detection studies was the R^2 score, which is commonly employed in regression research and establishes the capability of the model regression function which equals 9% (4/47). The percentages of the evaluation metrics used in the regression and classification investigations are shown in Figure 5. However, the ML techniques adopted by the selected articles were analysed, computed statistically and categorized into classification and regression as shown in figure 6a, and 6b.

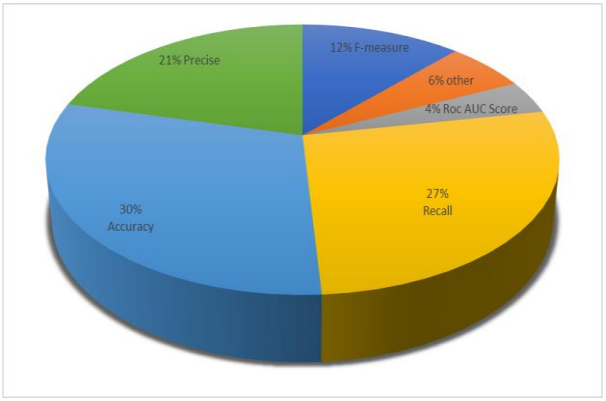


Fig. 5 (i): Frequency of evaluation matrix (Classification).

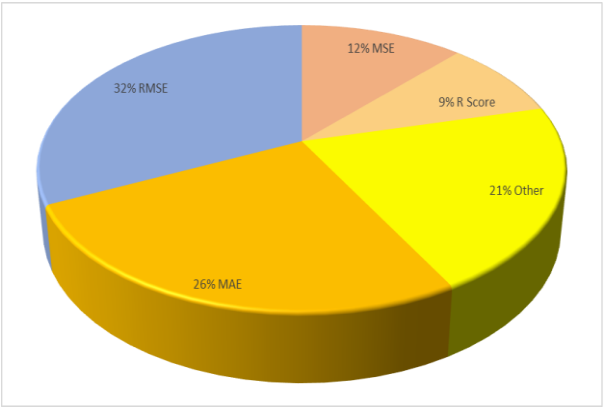
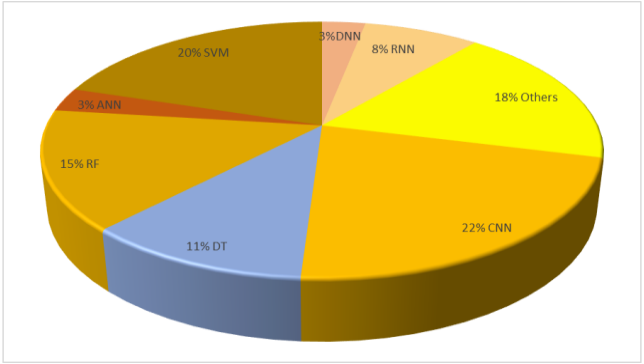
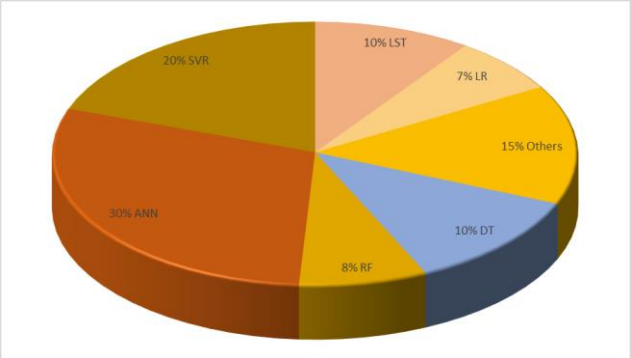


Figure 5 (ii): Frequency of evaluation matrix (Regression)



6a. Classification



6b. Regression

Figure 6 (a, b): ML Techniques in malware detection and Analysis.

4.5. Validation Techniques

Since training for both regression and classification tasks can be done in different ways, no distinction was considered when analyzing validation metrics in the study. The hold-out approach, which excludes data mining applications, was also used in the study, even though k-fold cross-validation is often utilized. 28% (64/228) of the studies used the hold-out approach, whereas 32% (73/228) used k-fold cross-validation to validate their findings. The frequency of the method that used data mining/training or data selection was 24% (55/228), however, studies for which the validation procedures were unknown was computed to be 16% (36/228). The percentages of the validation procedures considered in the experiments are displayed in Figure 7.

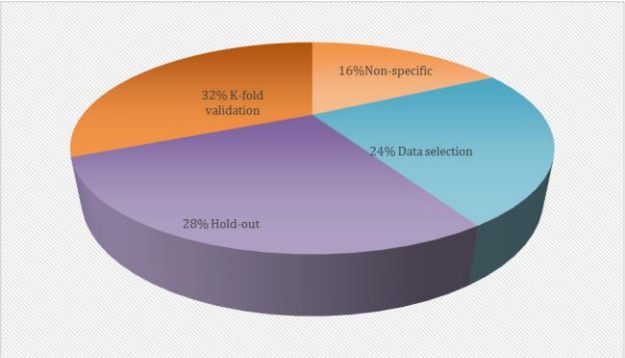


Figure 7: Validation Methods

4.6. Research Gap and Possible Remedies

There is a glaring gap in the studies that systematically assess the effectiveness of the various machine learning models for malware detection, even though a sizable body of literature has covered many aspects of malware detection and analysis. By exploiting an inherent asymmetry on these gaps, attackers might continuously breach the system's defenses in a manner that will be difficult to prevent (Johnson & Grumbling, 2019).

There aren't enough samples to train an ML model in most cases, which may result to high number of false positive due to limited number of instances in the sample. Defenders frequently place a high priority on lowering the number of false positives because they might be expensive. Nevertheless, this has the effect of increasing the rate of false negatives, which implicitly raises the possibility of a successful attack (Johnson & Grumbling, 2019). A hybrid machine learning model that combines two different models is capable of constructively detecting malware with miss-classification rate kept to a minimum.

Security tools are frequently used by parties other than the resource owners themselves and operate on several interdependent systems. Therefore, maintaining a low number of false positives without providing opportunities for attackers becomes a delicate issue (Johnson & Grumbling, 2019). However, a hybrid machine learning model uses a LinearSVC or interpolator with a smooth/linear structure that cannot be easily adjusted.

Most machine learning (ML)-based classification tools are accuracy-based, this implies that they determine whether an event is an attack based on probability. This is because machine learning tools for malware detection have not been updated to meet the requirements of offensive or defensive cyber operations (Zheng, et. al., 2022). It is worthy to note that the parameters of a hybrid ML systems for malware detection unlike the conventional ML tools can be upgraded and optimized to perform excellently without bias.

The glaring void inhibited because of limited access to malware dataset as well as obtaining more static and dynamic features for higher accuracy and detection (Singh & Jain, 2017). The lack of access to datasets for sophisticated malware samples, like metamorphic and polymorphic malwares, whose detection and classification may be effective than those commonly used by most researchers also pose a huge challenge. This glaring void and challenge have been tackled with the access to large updated benign and malware (metamorphic and polymorphic) dataset.

5. Conclusion

The use of ML in research has eventually top the chart in various field of studies including malware analysis and detection. The effectiveness of malware detection can be felt in terms of its goal, methodology, datasets, evaluation and validation techniques. Over 200 published articles were reviewed but it is difficult to apply these findings in practice due to variations. The features of the dataset and the objectives of the individual studies were connected to the use of classification and regression tasks in malware analysis and detection. To direct future research with potential solutions, this study sought to uncover the key distinctions, patterns, and issues in malware detection and analysis. First, a thorough assessment of the literature on ML-based malware analysis and detection was conducted. Secondly, the studies were categorized based on their objectives, models, assessment metrics, and validation techniques. Numerical data pertaining to the investigations were provided. It is recommended that k-fold cross-validation, a common validation method in research be considered for both problem domains (classification and regression) to assess the efficacy of every technique or framework that has been proposed. Future research should concentrate on deep learning, especially RNN, due to the advancement in AI and the growing population of users on the cyberspace.

References

Agarap, A. (2018). Towards building an intelligent anti-malware system: A deep learning approach using support vector machine (SVM) for malware classification. *arXiv*. https://doi.org/10.48550/ARXIV.1801.00318

Ambusaidi, A., Xiangjian, H., Nanda, P., & Tan, Z. (2016). Building an intrusion detection system using a filter-based feature selection algorithm. *IEEE Transactions on Computers*, 65(10), 2986–2998.

Azeez, N., Odufuwa, O., Misra, S., Oluranti, J., & Damaševicius, R. (2021). Windows for malware detection using ensemble learning. *Informatics*, 8(1). https://doi.org/10.3390/informatics8010010

Breiman, L. (2001). Random forests. *Machine Learning*, 45, 5-32. https://doi.org/10.1023/A:1010933404324

Bush, I.. & Abiyev, R. (2023). Introduction to machine learning and IoT. In *Machine learning and the internet of things in education* (Studies in Computational Intelligence, Vol. 1115, pp. 1-7). Springer. https://doi.org/10.1007/978-3-031-42924-8_1

Bush, I., Abiyev, R., Ma'aitah, M., & Altıparmak, H. (2018). Integrated artificial intelligence algorithm for skin detection. *ITM Web of Conferences*, 16, 02-24. https://doi.org/10.1051/itmconf/20181602024

Bush, I., Mansur, M., & Abubakar, U. (2023). Machine learning based cardless ATM using voice recognition techniques. In *Machine learning and the internet of things in education* (Studies in Computational Intelligence, Vol. 1115, pp. 75-84). Springer. https://doi.org/10.1007/978-3-031-42924-8_6

Caruana, G., Li, M., & Qi, M. (2011). A MapReduce based parallel SVM for large-scale spam filtering. In *Proceedings of the Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)* (pp. 26-30). IEEE.

Cen, L., Gates, S., Luo, S., & Li, N. (2015). A probabilistic discriminative model for android malware detection with decompiled source code. *IEEE Transactions on Dependable and Secure Computing*, 12(4), 400–412.

Chen, L.;Chen, P.; Lin,Z (2020) .Artificial Intelligence in Education: A Review. IEEE Access, 8,

75264–75278,
doi:10.1109/ACCESS.2020.2988510.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *arXiv*. https://doi.org/10.1145/2939672.2939785.

Damodaran, A., Fabio, T., Visaggio, C., Austin, T., & Stamp, M. (2017). A comparison of static, dynamic, and hybrid analysis for malware detection. *Journal of Computer Virology and Hacking Techniques*, 13(1), 51-66.

Dougherty, G. Pattern Recognition and Classification; Springer: Berlin/Heidelberg, Germany, 2013.

Duhan, N., Sharma, K., & Bhatia, K. (2009). Page ranking algorithms: A survey. In *Proceedings of the 2009 International Advance Computing Conference (IACC)* (pp. 1530–1537). http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4809246&isnumber=48 (Retrieved March 1, 2025).

Ever, Y.; Dimililer, K.; Sekeroglu, B. (2019). Comparison of Machine Learning Techniques for Prediction Problems; 2019; Advances in Intelligent Systems and Computing, 927. Springer, Cham doi: 10.1007/978-3-030-15035-8_69.

Gofwen, M., Idoko, B., JB Idoko: Application of Zero-Trust Networks in e-Health Internet of things (IoT) Deployment. Machine Learning and Internet of Things in Education: Models and Applications. Pp.209-233 © 2023 Springer Nature.

Helwan, A., Bush, I., & Abiyev, R. (2017). Machine learning techniques for classification of breast tissue. *Procedia Computer Science*, 120, 402–410.

Idoko, B., JB Idoko: IoT Security Vulnerability Assessment of E-learning Systems. Machine Learning and Internet of Things in Education: Models and Applications. Pp.235-243 © 2023 Springer Nature.

Idoko, B., JB Idoko, YZM Kazaure, YM Ibrahim, FA Akinsola, AR Raji. (2022): IoT Based Motion Detector Using Raspherry Pi Gadgetry. 2022 5th Information Technology for Education and Development (ITED), 1-5. 978-6654-9373-3/22 $31.00 (c) 2022 IEEE

Idoko, B. & Nwankwo, K. (2025). Development of a framework for cybersecurity Risk Assessment in the Maritime industry. Kwaghe International Journal of Engineering and Information Technology. doi:10.58578/kijeit.v2i2.5342

Johnson, A., & Grumbling, E. (2019). Implications of artificial intelligence for cybersecurity: Proceedings of a workshop. In *National Academies of Sciences, Engineering, and Medicine*. Washington, DC: https://doi.org/10.17226/25488.

Kannadhasan, S., Nagarajan, R., & Thenappan, S. (2022). Intrusion detection techniques based secured data sharing system for cloud computing using MSVM. In *9th International Conference on Computing for Sustainable Global Development (INDIACom)*. IEEE. https://doi.org/10.1109/INDIACom.2022.00000.

Khashman, A.; Carstea, C. Oil price prediction using a supervised neural network. Int. J. Oil Gas Coal Technol. 2019, 20, 360, doi:10.1504/IJOGCT.2019.098458.

Kim, T., Kang, B., Rho, M., Sezer, S., & Im, E. (2019). A multimodal deep learning method for Android malware detection using various features. *IEEE Transactions on Information Forensics and Security*, 14(3), 773–788. https://doi.org/10.1109/TIFS.2866319.

Liu, L., Wang, B., Yu, B., & Zhong, Q. (2017). Automatic malware classification and new malware detection using machine learning. *Frontiers of Information Technology and Electronic Engineering*, 18, 1336–1347. https://doi.org/10.1631/FITEE.1601325

Mahindru, A., & Sangal, A. (2020). MLDroid—Framework for Android malware detection using machine learning techniques. *Neural Computing and Applications*, 33(10), 5183–5240. https://doi.org/10.1007/s00521-020-05309-4

Maigida, A., Abdulhamid, S., Olalere, M., Alhassan, K., Chiroma, H., & Dada, E. (2019). Systematic literature review and metadata analysis of ransomware attacks and detection mechanisms. *Journal of Reliable Intelligent Environments*, 5, 67–89. https://doi.org/10.1007/s4860-019-00080-3

Mason, C., Twomey, J., Wright, D., & Whitman, L. (2018). Predicting engineering student attrition risk using a probabilistic neural network and comparing results with a backpropagation neural network and logistic regression. *Research in Higher Education*, 59, 382–400. https://doi.org/10.1007/s11162-017-9473-z

Moustafa, N., & HuJillSlay, J. (2019). A holistic review of network anomaly detection systems: A comprehensive survey. *Journal of Network and Computer Applications*, 128, 33–55.

Moustafa, N., HuJillSlay, J., & Creech, G. (2017). Novel geometric area analysis technique for anomaly detection using trapezoidal area estimation on large-scale networks. *IEEE Transactions on Big Data*, 65-81.

Otoum, S., Kantarci, B., & Mouftah, H. (2020). A novel ensemble method for advanced intrusion detection in wireless sensor networks. In *IEEE Conference on Automation Science and Engineering* (pp. 511-533). https://doi.org/10.1109/INDIACom.2020.00000

Oytun, M., Tinazci, C., Sekeroglu, B., Acikada, C., & Yavuz, H. (2020). Performance prediction and evaluation in female handball players using machine learning models. *IEEE Access*, 8, 116321–116335. https://doi.org/10.1109/3004182

Ozcil, I.; Esenyel, I.; Ilhan, A. A Fuzzy Approach Analysis of Halloumi Cheese in N. Cyprus. Food Anal. Methods 2021, doi:10.1007/s12161-021-02075-4.

Ozsahin, D., BB Duwa, Idoko, B. (2024) A Aleter, JB Idoko, I Ozsahin: Sleep Apnea Detection Device. Practical Design and Applications of Medical Devices. Pp 147-152. © 2024 Elsvier.

Pahlavan-Rad, M., Dahmardeh, K., Hadizadeh, M., Keykha, G., Mohammadnia, N., & Keikha, G. (2018). McBoost: Boosting scalability in malware collection and analysis using statistical classification of executables. In *Proceedings of the Annual Computer Security Application Conference (ACSAC)* (pp. 301–310). ACM Press.

Perrotta, C.; Selwyn, N. Deep learning goes to school: Toward a relational understanding of AI in education. Learn. Media Technol. 2019, 45, 1–19, doi:10.1080/ 17439884.2020.168 60

Rafique, M., Ali, M., Qureshi, A., Khan, A., & Mirza, A. (2019). Malware classification using deep learning-based feature extraction and wrapper-based feature selection technique. https://doi.org/10.48550/ARXIV.1910.10958

Salehi, Z., Ghiasi, M., & Sami, A. (2021). A miner for malware detection based on API function calls and their arguments. *16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP)* (pp. 563–568). IEEE. https://doi.org/10.1109/AISP.2021

Sekeroglu, B., Abiyev, R., Ilhan, A., Arslan, M., & Bush, I. (2021). Systematic literature review on machine learning and student performance prediction: Critical gaps and possible remedies. *Applied Sciences*, 11(22), 10907.

Sekeroglu, B., & Dimililer, K. (2020). Review and analysis of hidden neuron number effect of shallow back propagation neural networks. *Neural Network World*, 30, 97–112. https://doi.org/10.14311/NNW.2020.30.008

Sekeroglu, B., Dimililer, K., & Tuncal, K. (2019). Artificial intelligence in education: Application in student performance evaluation. *Dilemas Contemporaneos: Educacion, Politica y Valores*, 7(1), 1–21.

Sekeroglu, B.; Tuncal, K. (2021). Prediction of cancer incidence rates for the European continent using machine learning models. Health Inform. J., 27, 1460458220983878.

Selamat, N., & Ali, F. (2019). Comparison of malware detection techniques using machine learning algorithms. *Indonesian Journal of Electrical Engineering and Computer Science*, 16(1), 435–440. https://doi.org/10.11591/ijeecs.v16.i1.pp435-440

Singh, A., & Jain, A. (2017). Integrated malware analysis using machine learning. In *2nd International Conference on Telecommunication and Networks (TEL-NET)*. IEEE Xplore.

Uwanuakwa, I., Bush, I., Mbadike, E., Resatoglu, R., & Alaneme, G. (2022). Application of deep learning in structural health management of concrete structures. *Proceedings of the Institution of Civil Engineers-Bridge Engineering*, 1–8. Thomas Telford Ltd.

Venkatraman, S., Alazab, M., & Vinayakumar, R. (2019). A hybrid deep learning image-based analysis for effective malware detection. *Journal of Information Security and Applications*, 1–6.

Watson, R., Marnerides, K., Mauthe, A., & Hutchison, D. (2016). Malware detection in cloud computing infrastructures. *IEEE Transactions on Dependable and Secure Computing, 13*(2), 192–205.

Xu, Z., Ray, S., Subramanyan, P., & Malik, S. (2017). Malware detection using machine learning-based analysis of virtual memory access patterns. In *Design, Automation & Test in Europe Conference & Exhibition* (pp. 169–174). https://doi.org/10.23919/DATE.2017.7926977

Yang, L., Wu, H., Jin, X., Zheng, P., Hu, S., Xu, X., Yu, W., & Yan, J. (2020). Study of cardiovascular disease prediction model based on random forest in eastern China. *Scientific Reports, 10*, 1–8. https://doi.org/10.1038/s41598-020-62133-5

Zheng, M., Robbins, H., Chai, Z., Thapa, P., & Moore, T. (2022). Cybersecurity research datasets: Taxonomy and empirical analysis. In *Proceedings of the 11th USENIX Workshop on Cyber Security Experimentation and Test (CSET '22)*.

Zhong, W., & Gu, F. (2019). A multi-level deep learning system for malware detection. *Expert Systems with Applications, 133*, 151–162. https://doi.org/10.1016/j.eswa.2019.04.064