*Kufa Journal of Engineering*
*Vol. 16, No. 3, July 2025, P.P. 82-100*
*Article history: Received 17 May 2024, last revised 8 July 2024, accepted 20 July 2024*

# ENHANCING CYBERSECURITY THROUGH MALWARE DETECTION BASED ON MACHINE LEARNING TECHNIQUE

**Amer Mohamed Shhatha[1] and Omar Ibrahim Alsaif [2]**

**[1] Engineering Technical College/ Mosul, Northern Technical University, Mosul-Iraq,Email: amer.mohammed1@ntu.edu.iq**

**[2] Technical Engineering College For Computer and AI/ Northern Technical University, Mosul-Iraq, Email:omar.alsaif@ntu.edu.iq**

**https://doi.org/10.30572/2018/KJE/160306**

## ABSTRACT

The world is now more connected through technology and this has given rise to cyber threats, and malware is one of the threats that a system and data need to guard against. In this paper, we propose a detailed framework wherein state of the art ML methodologies can be employed for malware categorization and identification. In the evaluation of the performances of various ML algorithms, we analyze Random Forest, CatBoost, XGBoost, K-Nearest Neighbors (KNN), Histogram-based Gradient Boosting (Hist GB), and AdaBoost. The algorithms are assessed in this study using an assembly command dataset and a static and dynamic analysis approach to improve the detection rate and stability. Out of all algorithms discussed, Random Forest, CatBoost, XGBoost, Hist GB are ranked highest with 99% accuracy. As for the accuracy, KNN yielded an accuracy of 97%. Performance analysis based on metrics shows that Random Forest, CatBoost, and Hist GB not only have high accuracy but also precision, recall, and F1-score. Particularly, the accuracy of Random Forest was 99% for both the precision, recall, and F1-score. These results confirm the use of ML-based solutions in the analysis and counteraction of modern malware threats and their advantages over traditional detection methods as well as strengthening cybersecurity.

## KEYWORDS

Cybersecurity, Malware Detection, Machine Learning, Ensemble Methods, Classification, Cyber Threats.

## 1. INTRODUCTION

Malware can be shortly defined as malicious software, and it has been a problem that has never left people dealing with cybersecurity issues. It comprises a broad category of malicious programs such as viruses, worms, Trojan horses, ransomware, and spyware are part of the group. Malware can have maligned and cause massive data corruption and loss, financial loss, and disruption of service delivery. Previously used anti-virus programs include the signature based detection and heuristic analysis, which have constituted the major IT defense tools for quite some time now. However, the employments of these methods are not devoid of some demerits (Nikam, U.V. and Deshmuh, V.M., 2022). The major disadvantage of virus signature-based detection of the new type is that it is immobile against new, yet unknown, variety of the malware. Albeit the heuristic analysis is capable of detecting some samples of new and previously unknown malware, this analysis commonly provokes numerous false positives as well as false negatives thus making it rather untrustworthy (Akhtar, M. and Feng, T., 2022). Modern malware is versatile and diverse, which contributes to a very challenging battle in the detection phase. And the authors of malware are currently working hard to invent new ways of cloaking their code from previously successful detection methods. Mimicry has thus brought a desperate call for enhanced and more elastic methods of identifying malware (Darem, A.A., Ghaleb, F.A.,et al. 2021 ). Machine learning (ML) poses some solutions to these challenges by using the behaviors and pattern within a data to detect the malicious activities associated with malware, including the newer and emerging subtypes (Tao, F., Akhtar, M. and Jiayuan, Z., 2021).

Various tricksters utilize noxious programming consistently to take information, break organizations, or do monetary exchanges. Thus, safeguarding secret information has turned into a first concern for mainstream researchers. This work utilizes information mining and AI characterization methods to make an exhaustive system for distinguishing unsafe projects and shielding information from programmers (Chandrakala, D., et al). Our point is to give a powerful and viable strategy for malware grouping and identification by investigating highlights in view of the two marks and irregularities. Experimental examinations affirm that the proposed system is superior to different methodologies (Akhtar, M.S. and Feng, T., 2022).

The security of websites in the modern day is seriously threatened by the frequency and sophistication of modern malware. Different kinds of cyberattacks that occur in the digital world, or cyberspace, are depicted in Fig.1, (Kumar, R.R. et al., 2021). Malware is an example of this hazard since it is expressly made to do harm to computers or networks by spying on users or stealing their money. Interestingly, malware attacks now target medical devices,

industrial control systems, Internet of Things devices, and environmental sensors in addition to more conventional targets. Modern spyware is constantly changing, with new code and behaviors added on a regular basis, making detection difficult. As a result, conventional signature-based defenses have not been able to stop the spread of malware. A wider range of defensive actions becomes necessary to counter this action (Akhtar, M.S. and Feng, T., 2022).

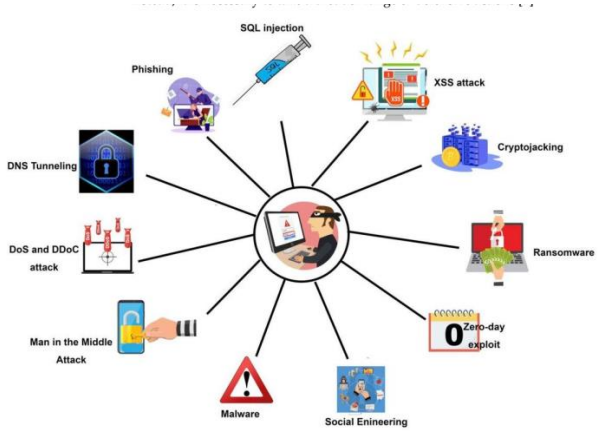


**Fig1. Types of cyberattacks (Akhtar, M.S. and Feng, T., 2022)**

It is essential to use both static and dynamic learning techniques to spot behavioural patterns that are common to all malware in a family. Static analysis examines potentially dangerous files' contents without running them, but dynamic analysis takes into account how they behave by monitoring function calls, following data flows, and incorporating monitoring code into dynamic binaries (Gibert, D.,et al, 2019). By utilizing both static and behavioural artefacts, machine learning algorithms are able to distinguish the changing architecture of modern malware, which makes it possible to detect sophisticated attacks that would elude conventional signature-based methods. Machine learning-based solutions are good at recognizing newly released malware, in contrast to signature-based approaches. Deep learning techniques improve feature representation accuracy by automatically engineering features (Firdaus, A., etal, 2018). Martin (2018) Digital Kill Chain, which is utilized for network security and cyberattack avoidance, is displayed in Fig.2. Amazon Web Services (AWS) opposed a huge dispersed forswearing of administration (DDoS) assault in February 2020, in spite of a (2.3) Tbps attack that brought a parcel sending pace of (293.1) Mbps and a solicitation pace of 694,201. This is supposedly one of the greatest DDoS goes after any point noticed. Three programmers accessed Twitter in July 2020 and assumed control over the records of notable clients, like Elon Musk of Tesla, Jeff Bezos of Amazon, and President Obama. Tricks including bitcoin that spread from hacked accounts made more than 100,000 $. Three individuals were then charged by the US Equity Division, one of whom was underage at that point (Akhtar, M. and Feng, T., 2022).

A cyberattack on Marriott's Starwood Lodgings in 2018 brought about the openness of more than 500 million clients' very own information. As per NHS Britain, at (2017) WannaCry ransomware attack harmed north of 300,000 frameworks across 150 nations and cost billions to fix as shown in Fig.2, (Akhtar, M. and Feng, T., 2022).


**Fig 2. Martin Cyber Kill Chain for prevention of cyber intrusions activity**

In 2017, Russia launched a cyberattack on the electrical infrastructure of Ukraine in an attempt to weaken its neighbours. With this action, Russia made its foray into the realm of large-scale cyberwarfare and proved its mettle. Even though this sophisticated strike happened a year after Russia annexed Crimea, which started its war with Ukraine, it was the first successful attack on electricity infrastructure. The attack on the command centre was led by the Russian cyber military outfit Sandworm. Hackers took control of the computer systems by taking advantage of holes in the command center's defences, which caused the substation to shut down. The effect was exacerbated by additional attacks that went after other substations. The attack is thought to have affected between 200,000 and 300,000 individuals in total (Pavithra, J. and Femilda Josephin, J.S., 2020). The reason for applying machine learning in the detection of malicious software is as a result of the data processing capacity of the machine learning especially in processing the data and coming up with better detection results in future. While most previous approaches can merely recognize the pattern presented in the training set and fail to detect new strains of malware, ML algorithms have the ability to extend this knowledge to recognize new similar strains of malware, making cybersecurity solutions stronger and more efficient. Nevertheless, significant chasms in the current status of research and deployment of the ML in the context of malware detection exist. A lot of research has been done within a specific category of algorithms or in small sets of data, which created issues with the relevance of these techniques to solve real-life problems.

This paper will fulfill these shortcomings by offering a total efficiency assessment of several state-of-the-art machine learning algorithms that include Random Forest, CatBoost, XGBoost,

K-Nearest Neighbors (KNN), Histogram-based Gradient Boosting (Hist GB) and AdaBoost. For the static analysis, we use an extensive assembly command dataset while for dynamic analysis, we provide a more diverse range of binaries for testing these algorithms. This paper aims at investigating the performance of various types of ML methodologies in the classification of malware and go further in recommending the most pertinent ones for the same with detailed performance outcomes alongside with the merits as well as demerits of each method. In so doing, the proceedings of this workshop will help to further immediately related literature and idea, provide appropriate proposal and guidelines for enhanced malware detection, and thus help create stronger cybersecurity systems.

## 2. RELATED WORKS

The reliability and uprightness of organized frameworks are genuinely compromised by assaults known as disseminated refusal of administration (DDoS). Quick assault discovery and alleviation is important to keep up with the trustworthiness and security of online administrations. Throughout the long term, scientists have fostered different strategies and procedures focused on successfully distinguishing and moderating DDoS assaults. This part gives a rundown of relevant papers on the subject of DDoS assault identification.

Khalaf et al. (2019): Played out a careful investigation of DDoS assault identification techniques utilizing a scope of computer-based intelligence and factual philosophies. nitty gritty various techniques yet discarded data about the specific datasets that were utilized (Khalaf, B.A., et al, 2019).

The effect of misleading up-sides or negatives was not tended to by (Hosseini and Azizi 2019) in their cross breed structure for identifying fast DDoS assaults that utilizes many AI approaches (Wani et al. 2019). In spite of the absence of data in regards to the dataset's qualities, they found exorbitant DDoS assaults in a cloud climate with phenomenal precision rates. (Alsirhani, Sampalli, and Bodorik 2019) introduced a DDoS identification framework involving characterization strategies and fluffy rationale in Apache Flash, underlining versatility however giving no insights concerning the fluffy rationale system (Alsirhani, A., Sampalli, S. and Bodorik, P., 2019 ). (Shamsolmoali and Zareapoor,2014) offered a measurable procedure for dependably recognizing and arranging TCP DDoS assaults, but they gave little data on different kinds of DDoS assaults (Xiao, P.et al, 2015), presented a CKNN-based method for DDoS detection that used link analysis; although it lacked dataset details, it achieved better accuracy (Kuang, F., et al, 2015), which. proposed a support vector device-based approach that improved performance but lacked information on the use of chaotic particles and datasets. Zekri et al.

(2018) Provided a hybrid technique that integrated many detection approaches, but lacked information on detection accuracy, for known and unknown attack detection utilising Snort and C4.5. (Kushwah and Ali, 2018), proposed a model for DDoS discovery utilizing ANN prepared with dark opening streamlining calculations, using improvement for ANN preparing however missing subtleties on design and preparing process. In a distributed computing setting, (Kushwah and Ranga, 2020) fostered a DDoS discovery framework using V-ELM and contrasted it with other ML calculations; nonetheless, the dataset highlights were not uncovered. Table 1 below Discuss these works.

**Table 1. Related Works Analysis**

| Study | Methodology | Strengths | Limitations | Key Contributions |
|---|---|---|---|---|
| Khalaf et al. (2019) | AI and statistical methodologies | Comprehensive analysis of various techniques. | No details of the two datasets are provided | Emphasized different and most utilized AI prevention |
| Hosseini and Azizi (2019) | The proposed hybrid framework involves using more than one way of working with ML | Of course, the work can and should be continued, using innovative forms of hybrid ML approaches. | The possibility of false positives/negatives was not considered as part of its approach. | Introduced a hybrid solution in the context of fast DDoS detection |
| Wani et al. (2019) | Detecting in cloud-based environment | Achieved excellent accuracy rates | No information or inadequate information about dataset tells you. | Showed capability of distinguishing DDoS attacks |
| Alsirhani, Sampalli, and Bodorik (2019) | Classification techniques and fuzzy logic within Apache Spark | Emphasized scalability | has not provided details on the mechanism of fuzzy logic in his paper. | Suggested the design of a flexible DDoS s. |
| Xiao et al. (2015) | Grocery-only path-based method and CKNN- | Achieved better accuracy | Lack of dataset details | proposed CKNN-based |
| Zekri et al. (2018) | Totally new for vehicle detection, | Combined Snort and C4. Increasing the level of machine | The care provider was not informed of the accuracy of detection in this type of cancer. | Combined the multiple detection techniques |
| Kushwah and Ranga (2020) | V-ELM used in a cloud computing environment | As it stands, some other ML algorithms are diverse but differ in terms of complexity and effectiveness across datasets. | No dataset feature information | Deployed a DDoS detection system based on the V-ELM algorithm in cloud environments |

Some of the strengths found in this study that uses the surveys include: The following are limitations found in this study that uses the surveys Among the strengths that apply in this study that uses surveys include: This study extends the findings of the previous works and overcome their weaknesses. And in order to improve the efficiency of DDoS detection, we will describe

a comparison of machine learning approaches like Random Forest, CatBoost, XGBoost, KNN, Hist GB, and AdaBoost in detail. For our experiment, we leverage a large assembly command dataset for static and dynamic analysis to get a comprehensive evaluation of the algorithms. By providing extensive descriptions of the features and properties of the datasets, the methods and approaches used in the detection of DDoS attacks and the evaluation metrics that we have employed such as accuracy, precision and recall, the work sheds more light on the best methodologies that can be used for the same. Not only is this study an extension and expansion of prior literature, but it is also empowering policy makers with tangible recommendations for bolstering defences against DDoS threats (Kushwah and Ranga.2020).

## 3. METHODOLOGY

The block diagram in Fig.3 shows a careful strategy for distinguishing malware utilizing AI methods. The malware dataset is first made by social occasion, cleaning, and preprocessing information. To help with the preparation and evaluation of the model, the dataset is then separated into preparing (80%) and testing (20%) sets. An assortment of AI strategies, including Irregular Timberland, CatBoost, XGBoost, K-Nearest Neighbors (KNN), Histogram Gradiant Boosting (Hist GB), and AdaBoost, are then picked for preparing utilizing the preparation dataset. Utilizing the test dataset, the models' presentation is surveyed following preparation utilizing assessment measurements like as Accuracy, precision, recall, and F1-score. At last, the consequences of every calculation's assessment measurements are shown or printed, empowering an examination of the calculations' viability in distinguishing malware.

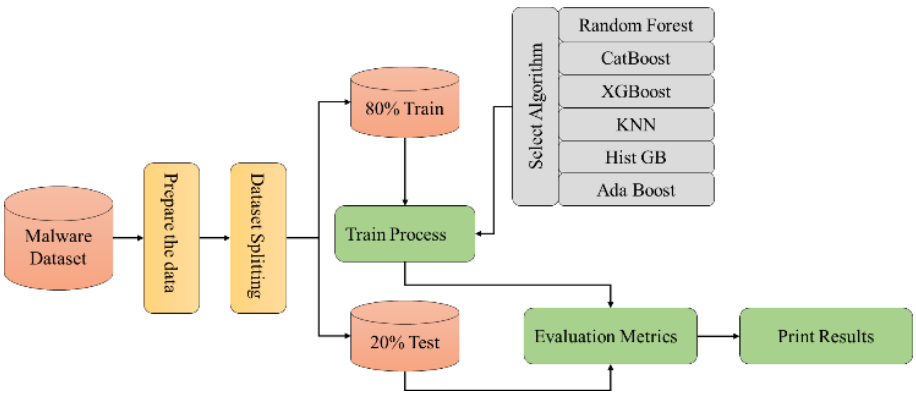


**Fig. 3. Model Block Diagram**

### 3.1. Environment and Libraries

Google Colab or Collaboratory is a free cloud based Python notebook which provides users the access to powerful natives and additionally allowing to execute computations within GPUs and TPUs without having these resources on local computer. It also includes popular machine learning libraries like scikit-learn, XGBoost, Catboost and others, makes it suitable to

implement and test purpose of machine learning algorithms like, Random Forest, Catboost, XGBoost, K-Nearest Neighbor (KNN), Hist Gradient boosting (Hist GB), AdaBoost etc. Such libraries can be integrated well with Colab and helps in building models for various malware detection tasks at a fast pace. Moreover, since code can be run in online environments, it is easier to work on projects remotely, or helps in effective collaboration, especially when using versioning and real-time editing, which is perfect for research and educational purposes.

### 3.2. Dataset Description

The features related to assembly (ASM) commands, together with two additional columns ("line_count_asm" and "size_asm") and a target variable ("Class"), are sourced from Kaggle (https://www.kaggle.com/datasets/dscclass/malware/data"). The ASM commands contain a large number of instructions, such as addition (add), call, compare (cmp), jump (jmp), move (mov), pop, push, and many more. Each instruction is represented as a separate feature. Additional ASM capabilities include the ability to display sizes, line counts, and command counts. It seems that the target variable "Class" divides instances into many groups or categories (Rashid, S.J., Baker, S.A., et al 2024). The dataset appears well-structured and has the potential to be utilised for training machine learning models for classification tasks, particularly in the context of cybersecurity applications like virus detection. The Maleware classes are:

1. Trojan Horse: Malware that poses as trustworthy software and is frequently used to access systems without authorization.
2. Worm: Malware that multiplies and travels over networks, using up bandwidth and slowing down computers.
3. Virus: Malware that affixes itself to trustworthy programs and disseminates itself upon execution of the compromised program.
4. Ransomware: Malware that scrambles information on a PC and solicitations cash to unscramble it.
5. Spyware: Malware that clandestinely gathers information about a client's activities without the client's consent.
6. Adware: Malware that covers up a client's contraption with meddlesome adverts.
7. Rootkit: Malware that covers its presence and permits unapproved clients to get to a framework.
8. Botnet: Malware that contaminates various gadgets and changes them into "bots" under the noxious control of a focal server.
9. Keylogger: Malware that logs keystrokes, giving programmers admittance to private information like Visa numbers and passwords.

### 3.3. Dataset Preprocessing

A fundamental stage in preparing information for AI models is dataset planning. The example code gave loads a dataset called "malware.csv" utilizing pandas, where 'X' represents highlights and 'y' for the objective variable, the 'Class' segment. The train_test_split capability from scikit-learn is then used to partition the dataset into preparing and testing sets, with an irregular state set for reproducibility and a test size of 20%.

Preprocessing methodology ordinarily include taking care of exceptions, scaling highlights, encoding clear cut factors, and dealing with missing qualities. Nonetheless, apparently these preliminary systems are not completed expressly founded on the code gave. It's memorable's pivotal that the sort and need of preprocessing stages can change in view of the necessities of the AI calculation being utilized as well as the highlights of the dataset. Consequently, prior to preparing a model on this dataset, extra preprocessing steps like overseeing missing qualities or encoding clear cut factors may be required. It's likewise really smart to explore the dataset in more detail to find out about its properties and recognize the appropriate preprocessing activities expected for the most ideal model presentation.

The data cleaning process focused on several important steps that helped to optimize the data for further machine's learning usage. First, noise reduction was performed in order to remove the features which could be less important and irrelevant to making a decision about a specific malware. For noise removal of the features, different feature selection methods were used to filter out the irrelevant features (Baker, S.A., et al, 2025). The next step which was followed in order to maintain a high level of uniformity in the nature of the models was to both normalize as well as standardize the data. Regarding the Miscellaneous features, the scaling of numerical features min-max scaling was done in the range 0 to 1, in addition, the features were standardized with a mean of 0 and standard deviation of 1. This step was more crucial for feature scaling as algorithms that are sensitive to feature scaling were performed before this step such as K-Nearest Neighbors (KNN) and the gradient boosting. Further, there was data preprocessing in handling the missing values in the numerical attributes by replacing the missing value with the average or median of the feature values and missing categorical attributes replaced with the most frequent value of the feature. These procedures of data cleaning seemed critical for cleaning up the data for use in training as well as testing the model and are explained here in detail in order to guarantee that any findings reproduced in the future are consistent with those presented here. Table 2 containing information regarding the computational assets for training and testing of the machine learning models.

**Table 2. Information Regarding the Computational Assets for Training and Testing of The Machine Learning Models**

| Hardware | Description | Software Environment | Description |
|---|---|---|---|
| CPU | 2x Intel Xeon @ 2.30GHz (2 cores) | Operating System | Ubuntu 18.04.5 LTS |
| GPU | NVIDIA Tesla K80, Tesla T4, P4, P100, or V100 GPU | Python Version | 3.7.x or 3.6.x (varies) |
| RAM | Up to 25GB RAM (Colab Pro) | Libraries/Frameworks | Various libraries pre-installed, including TensorFlow, PyTorch, Scikit-learn, Keras, Pandas, NumPy, Matplotlib, Seaborn, and others. |

### 3.4. Machine Learning Algorithms

Choosing the machine learning algorithms in this study was based on their theoretical advantages and disadvantages where they were applied, specifically in malware detection. Random Forest is a collection of decision trees that brings high noise tolerance and good pattern recognition capability but is computation expensive. CatBoost is a gradient boosting algorithm capable of handling categorical features and has a low risk of overfitting; however, the model is relatively slow in training. XGBoost is known to be effective with large datasets and high-dimensional data, although it is sensitive to the parameter setting. K-Nearest Neighbors (KNN) results in classification based on distance from pre-labeled samples; they are good for different distributions, but may take time and can be influenced by noisy attributes. Finally, AdaBoost increases the weak classifiers' performance through weight adjustment based on the mistakes made during classification; it performs well in difficult problems but is prone to noise and outliers. These algorithms were selected to offer a versatility to address the various problems that are associated with malware detection (Kushwah, G.S. and Ranga, V., 2020).

1. Random Forest: As a feature of its gathering growing experience, Irregular Woodland fabricates a few choice trees during preparing and yields the mean expectation (relapse) or the method of the classes (characterization) for each tree. By bringing down overfitting and helping power by consolidating a few trees, it beats the choice tree technique.

2. CatBoost: is a slope supporting method created by Yandex that performs uncommonly well with clear cut information. The treatment of all out factors is programmed, nullifying the requirement for impressive information preprocessing. CatBoost involves an arranged helping calculation related to a symmetric tree geography to save preparing time and increment execution (Genuer, R., Poggi, J.-M., and Villa-Vialaneix, N., 2015).

3. XGBoost: Outrageous Slope Supporting, or XGBoost, is another inclination helping strategy that has gained notoriety for effectiveness and execution. It utilizes a more regularized model

formalization to forestall overfitting, making it less inclined to overfitting than conventional inclination supporting. XGBoost is in many cases utilized in AI rivalries because of its standing for speed and exactness (Zhang, Y., Zhao, Z. and Zheng, J., 2020).

4. KNN (K-Nearest Neighbors): is a straightforward yet effective supervised machine learning algorithm used for classification and regression tasks. It predicts the class or value of a data point by considering the majority class or averaging the values of its K nearest neighbors, determined by a chosen distance metric such as Euclidean or Manhattan distance. While simple to implement, KNN's performance relies on selecting the appropriate K value and distance metric, and it can be computationally intensive for large datasets due to its need to calculate distances for each new data point. (Balli, O., 2022).

5. Hist GB (Histogram-based Angle Supporting): HistGB is a variation of inclination supporting calculations that uses histograms to proficiently register the slope measurements during the preparation interaction. This procedure is especially helpful for dealing with enormous datasets on the grounds that it lessens the computational expense of ascertaining angles contrasted with conventional slope supporting strategies. By binning the element values into histograms, HistGB can accelerate the preparation interaction while keeping up with or in any event, working on prescient exactness. It's a strong methodology frequently utilized in AI errands, particularly while managing enormous scope datasets (Vahedifar, M.A., Akhtarshenas, A., Sabbaghian, M. and Rafatpanah, M., 2023).

6. AdaBoost (Versatile Supporting): AdaBoost is a gathering learning technique that frames serious areas of strength for a by joining a couple of frail understudies, (for instance, decision trees). Each getting ready test is given a weight, and these heaps are then iteratively adjusted to highlight the erroneously characterized models in the going with cycles. AdaBoost can be sensitive to uproarious data and peculiarities, and it works honorably with delicate students (Guryanov, A., 2019) . The chosen algorithms are such that they provide a good balance between accurate, flexible, efficient and scalable when dealing with large large complex datasets of malwares. These avoid the aspect of any single solution lacking the other's ability in tackling malware since every solution offers a distinct way and means of approaching the problem. However, other algorithms were disregarded since certain algorithms present tradeoffs in performance, computational efficiency, and handling of the features of our dataset. The selection method has been done with a lot of precautions to guarantee that the study employs the relevant tools for optimum and credible malware detection. Table 3 describe training parameters.

**Table 3. Training Parameters**

| Algorithm | Training Parameters |
|---|---|
| Random Forest | n_estimators: 100, max_depth: None, min_samples_split: 2, min_samples_leaf: 1, max_features: 'sqrt' |
| CatBoost | iterations: 1000, learning_rate: 0.03, depth: 6, l2_leaf_reg: 3, random_strength: 1 |
| XGBoost | n_estimators: 1000, learning_rate: 0.1, max_depth: 6, subsample: 0.8, colsample_bytree: 0.8 |
| K-Nearest Neighbors (KNN) | n_neighbors: 5, weights: 'uniform', algorithm: 'auto', leaf_size: 30, p: 2 |
| HistGradientBoosting (Hist GB) | max_iter: 100, learning_rate: 0.1, max_leaf_nodes: 31, max_depth: None, min_samples_leaf: 20 |
| AdaBoost | n_estimators: 50, learning_rate: 1.0, algorithm: 'SAMME.R' |

### 3.5. Evaluation of Metrics

With regards to the article, the assessment measurements — Exactness, Accuracy, Review, and F1-Score — act as urgent measures to evaluate the presentation of an AI or grouping model. A disarray lattice goes with them to figure out the model's way of behaving (Shams, P. and Javaid, N., 2022) comprehensively. Accuracy: A key evaluation indicator called accuracy counts the percentage of cases that are correctly classified out of all the instances. It gives a general idea of how well the model is in predicting things. A model that performs better is indicated by higher accuracy numbers, although more is required when working with imbalanced datasets.

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

Precision: underlines the model's ability for exact, playful guaging. The proportion of genuine encouraging points to the absolute of misleading up-sides and genuine negatives is utilized to process it. At the point when misleading positive errors are costly, a high precision score proposes that the model is less inclined to deliver them, which can be significant.

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

Recall: which is a proportion of the model's precision in recognizing each sure case, is likewise alluded to as Responsiveness or Genuine Positive Rate. The proportion of genuine up-sides of the all out of obvious up-sides and bogus negatives is utilized to compute it. While missing positive cases is basic, high review is important.

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

F1-Score: A consonant mean of review and accuracy is the F1-Score. It gives a score that considers bogus up-sides and misleading negatives by adjusting these two models. The F1-Score is particularly useful when you want an exhaustive evaluation that considers accuracy and memory simultaneously.

$$1 - \text{Score} = 2 \times \frac{\text{percision} \times \text{recall}}{\text{Percison+recall}} \quad (4)$$

Confusion Matrix: A model's performance is visually represented by the confusion matrix, particularly in binary classification. The predictions of the model are divided into four groups: false positives, false negatives, True positives, and true negatives. It provides a thorough grasp of the areas in which the model performs well and those in which its predictions fall short (Vujović, Ž., 2021).

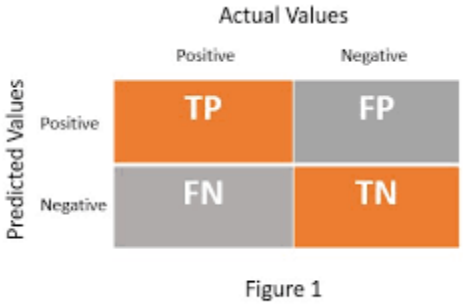


**Fig. 4. Confusion Matrix**

From the above analysis, the manuscript used accuracy, precision, recall, and F1-score as performance metrics essential in the current landscape of cybersecurity, with each metric serving unique purposes in evaluating machine learning models in malware detection. Accuracy reveals how many of the total number of instances were classified correctly on average in the case of a model. Accuracy on the other hand measures the predictive capability of the classifier in relation to its ability to identify positive instances and exclude negative instances and it measures the percent of positive cases which had been predicted out of all the cases that we have predicted to be positive. Common to cybersecurity is the problem of false positives, which refer to situations that cause the system to generate alarm or take some action when there is none to be funded. Sensitivity is mainly the ratio of correctly identified positive cases by the model to all the true positive cases which stresses on the importance of identifying all the malicious samples which can pose a threat. False negatives for malware detection could lead to unchecked and uncontrolled Malware and thus be a major threat to the product. Our evaluation metrics include precision that indicates the proportion of actual positives out of the total number of predictions, recall that shows the ratio of correctly identified positives relative to all actual positives, and the F-measure that captures the intermediate value of both precision and recall and thus avoids the domination of one of the two measurements. Due to the critical impact of false positive and false negative results in malware detection, metrics selection is critical to fairly assess the approximate model performance and establish effective cybersecurity strategies.

### 3.6. Web Application Design

Here's an example of a web page made with the Anvil platform that has links that point to malicious content. Potential victims receive these URLs over email, at which point they download the program. Victims test the software by downloading it and utilizing machine learning models to anticipate their states or behaviors (See Fig. 5).


**Fig. 5. Designed Anvil Web Application**

Considerable ethical and cybersecurity problems are brought up by this scenario. Spreading malware on purpose is against the law and unethical. Furthermore, it may be possible to forecast malware behaviour maliciously using machine learning models.

The significance of cybersecurity awareness and precautions must be emphasized. In order to safeguard their systems, users should be aware of the risks involved in downloading files from unidentified or dubious sources and utilize reliable antivirus software.

Additionally, following ethical rules is crucial when using machine learning models. Instead of being applied to activities that could endanger people or organizations, machine learning should be positive and advantageous. It is imperative that you notify the relevant authorities of any harmful behaviour you come across or suspect so that it can be investigated and mitigated.

## 4. RESULTS

The presentation boundaries, like exactness, accuracy, review, and F1-score, for each AI strategy evaluated on a specific dataset are shown in Fig.5 and Table 4.

Taking everything into account, the models perform splendidly, with exactness appraisals going from 0.97 to 0.99. This proposes that the models are effective in precisely sorting dataset cases. Besides, every calculation reliably accomplishes high accuracy, review, and F1-scores, proposing that notwithstanding their general exactness, these calculations likewise perform well

in precisely distinguishing positive cases (accuracy), catching every positive example (review), and finding some kind of harmony among accuracy and review (F1-score).

All models work perfectly, with exactness, accuracy, review, and F1-score scores of 0.99 for Random Forest, CatBoost, XGBoost, and Hist GB. These troupe learning procedures create tough and exceptionally precise models by using the upsides of angle supporting and various choice trees. The exactness, accuracy, review, and F1-score of the KNN model are all at 0.97, showing great execution, notwithstanding its little underperformance in contrast with the group draws near. KNN is famous for its effortlessness and interpretability and relies upon the likeness of models in the element space.

**Table 4. Metrics Analysis**

| Algorithm | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| Random Forest | 0.99 | 0.99 | 0.99 | 0.99 |
| CatBoost | 0.98 | 0.97 | 0.98 | 0.98 |
| XGBoost | 0.97 | 0.97 | 0.96 | 0.95 |
| KNN | 0.98 | 0.96 | 0.98 | 0.97 |
| Hist GB | 0.97 | 0.98 | 0.97 | 0.98 |

These outcomes show that the surveyed calculations perform well and are all hearty on the gave dataset, making them generally fitting choices for the main job. Eventually, different contemplations like interpretability, figuring effectiveness, and execution straightforwardness might impact the model choice.

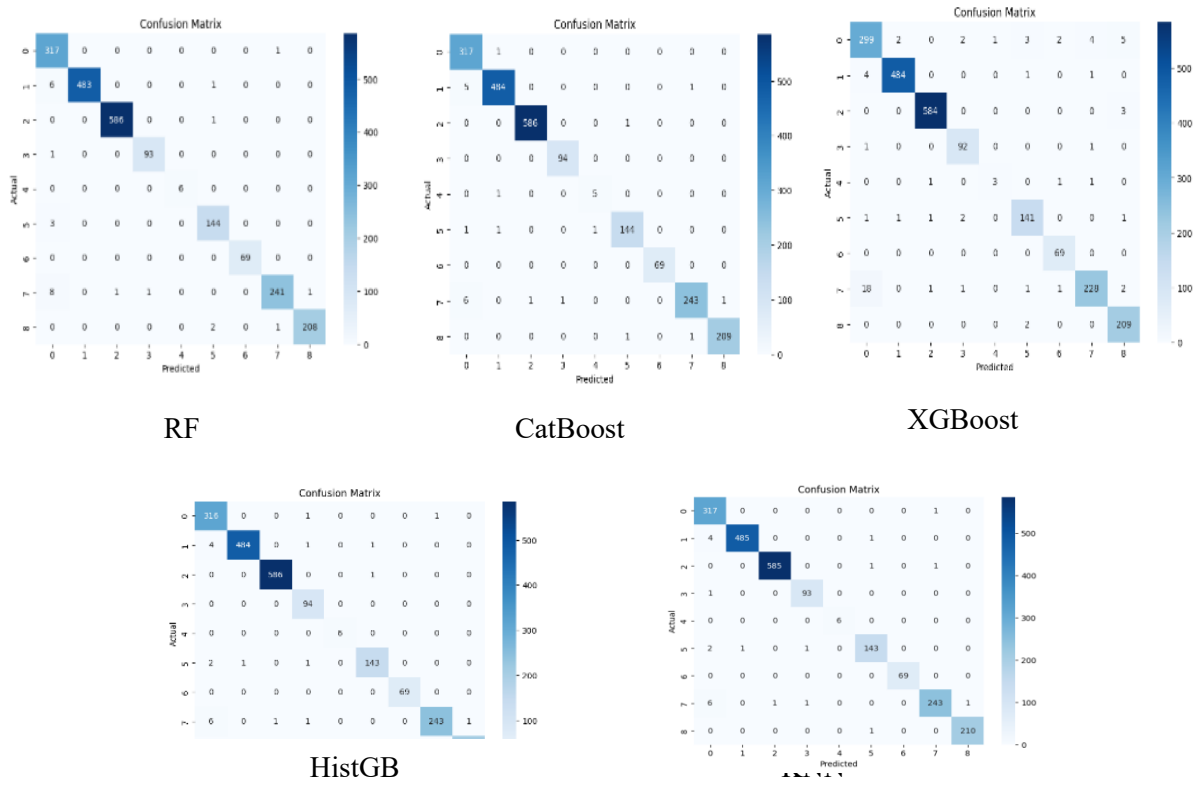


**Fig.6. Confusion matrix**

## 5. CONCLUSIONS

The exploration shows the viability of AI calculations in malware characterization and location, with group approaches beating more customary methods. Our strategy uses both static and dynamic data to recognize pernicious projects with high exactness and power. The assessed calculations — Irregular Woods, CatBoost, XGBoost, KNN, Hist GB, and AdaBoost — show solid execution, exhibiting their appropriateness for online protection applications. Future investigations could investigate bigger datasets and further developed strategies to further develop recognition abilities and address advancing digital dangers. In the domain of future works, specialists can investigate interdisciplinary coordinated efforts to address complex difficulties, focus on moral contemplations in mechanical headways, and spotlight on maintainability endeavors, for example, environmentally friendly power and environmental change relief. Furthermore, headways in information protection and security, alongside imaginative ways to deal with schooling and medical care conveyance, hold guarantee. Metropolitan arranging drives can be upgraded through savvy city innovations, while an emphasis on friendly effect and local area commitment stays pivotal for comprehensive advancement.

**Acknowledgment**

The authors would like to thank Northern Technical University for support.

## 6. REFERENCES

Ahmed, A.I., Khidhir, A.M., Baker, S.A., Alsaif, O.I., Saleh, I.A. 2024 “Enhancing Cybersecurity by relying on a Botnet Attack Tracking Model using Harris Hawks Optimization”, International Journal of Computers and their Applications, , 31(2), pp. 103–110

Akhtar, M. and Feng, T., 2022. IOTA based anomaly detection machine learning in mobile sensing. EAI Endorsed Transactions on Creative Technologies, 9(30), p.172814. doi: 10.4108/eai.11-1-2022.172814.

Akhtar, M.S. and Feng, T., 2022. Detection of sleep paralysis by using IoT based device and its relationship between sleep paralysis and sleep quality. EAI Endorsed Transactions on Internet of Things, 8(30), p.e4. doi: 10.4108/eetiot.v8i30.2688.

Akhtar, M.S. and Feng, T., 2022. Malware analysis and detection using machine learning algorithms. Symmetry, 14(11). doi: 10.3390/sym14112304.

Alsirhani, A., Sampalli, S. and Bodorik, P., 2019. DDoS detection system: Using a set of classification algorithms controlled by fuzzy logic system in Apache Spark. IEEE Transactions on Network and Service Management, PP(c), p.1. doi: 10.1109/TNSM.2019.2929425.

Baker, S.A., Thanoon, K.H., Abed, M.N., ... Abdulqader, T.Y., Alsaif, O.I., 2025 "A Secure and Privacy-Centric Blockchain Platform for Cloud-Based Healthcare Data Management", 2nd International Conference on IT Innovations and Knowledge Discovery ITIKD.

Balli, O., 2022. Use of XGBoost algorithm in classification of EEG signals.

Chandrakala, D., Sait, A., Kiruthika, J. and Nivetha, R., 2021. Detection and classification of malware. In: 2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA). pp.1–3. doi: 10.1109/ICAECA52838.2021.9675792.

Darem, A.A., Ghaleb, F.A., Al-Hashmi, A.A., Abawajy, J.H., Alanazi, S.M. and Al-Rezami, A.Y., 2021. An adaptive behavioral-based incremental batch learning malware variants detection model using concept drift detection and sequential deep learning. IEEE Access, 9, pp.97180–97196. doi: 10.1109/ACCESS.2021.3093366.

Firdaus, A., Anuar, N.B., Karim, A. and Razak, M.F.A., 2018. Discovering optimal features using static analysis and a genetic search based method for Android malware detection. Frontiers of Information Technology & Electronic Engineering, 19(6), pp.712–736. doi: 10.1631/FITEE.1601491.

Genuer, R., Poggi, J.-M., Tuleau-Malot, C. and Villa-Vialaneix, N., 2015. Random forests for big data. Big Data Research, 9, Nov. doi: 10.1016/j.bdr.2017.07.003.

Gibert, D., Mateu, C., Planes, J. and Vicens, R., 2019. Using convolutional neural networks for classification of malware represented as images. Journal of Computer Virology and Hacking Techniques, 15(1), pp.15–28. doi: 10.1007/s11416-018-0323-0.

Guryanov, A., 2019. Histogram-based algorithm for building gradient boosting ensembles of piecewise linear decision trees. In: Proceedings of the International Conference on Machine Learning and Data Mining. pp.39–50. doi: 10.1007/978-3-030-37334-4_4.

Hosseini, S. and Azizi, M., 2019. The hybrid technique for DDoS detection with supervised learning algorithms. Computer Networks, 158, pp.35–45. doi: 10.1016/j.comnet.2019.04.027.

Khalaf, B.A., Mostafa, S.A., Mustapha, A., Mohammed, M.A. and Abduallah, W\.M., 2019. Comprehensive review of artificial intelligence and statistical approaches in distributed denial

of service attack and defense methods. IEEE Access, 7, pp.51691–51713. doi: 10.1109/ACCESS.2019.2908998.

Kuang, F., Zhang, S., Jin, Z. and Xu, W., 2015. A novel SVM by combining kernel principal component analysis and improved chaotic particle swarm optimization for intrusion detection. Soft Computing, 19(5), pp.1187–1199. doi: 10.1007/s00500-014-1332-7.

Kumar, R.R. et al., 2021. Report on the follow-up to the regional implementation strategy of the Madrid international plan of action on ageing in Lithuania. Frontiers in Neuroscience, 14(1), pp.1–13.

Kushwah, G.S. and Ali, S.T., 2018. Detecting DDoS attacks in cloud computing using ANN and black hole optimization. In: 2nd International Conference on Telecommunication Networks (TEL-NET). pp.1–5. doi: 10.1109/TEL-NET.2017.8343555.

Kushwah, G.S. and Ranga, V., 2020. Voting extreme learning machine based distributed denial of service attack detection in cloud computing. Journal of Information Security and Applications, 53. doi: 10.1016/j.jisa.2020.102532.

Nikam, U.V. and Deshmuh, V.M., 2022. Performance evaluation of machine learning classifiers in malware detection. In: 2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE). pp.1–5. doi: 10.1109/ICDCECE53908.2022.9793102

Pavithra, J. and Femilda Josephin, J.S., 2020. Analyzing various machine learning algorithms for the classification of malwares. IOP Conference Series: Materials Science and Engineering, 993(1). doi: 10.1088/1757-899X/993/1/012099.

Rashid, S.J., Baker, S.A., Alsaif, O.I., Ahmad, A.I., 2024 “Detecting Remote Access Trojan (RAT) Attacks based on Different LAN Analysis Methods”, Engineering Technology and Applied Science Research, , 14(5), pp. 17294–17301.

Shams, P. and Javaid, N., 2022. Adaptive boosting (AdaBoost) algorithm.

Shamsolmoali, P. and Zareapoor, M., 2014. Statistical-based filtering system against DDoS attacks in cloud computing. In: 2014 International Conference on Advanced Computing and Communication Informatics (ICACCI). pp.1234–1239. doi: 10.1109/ICACCI.2014.6968282.

Tao, F., Akhtar, M. and Jiayuan, Z., 2021. The future of artificial intelligence in cybersecurity: A comprehensive survey. EAI Endorsed Transactions on Creative Technologies, 8(28), p.170285. doi: 10.4108/eai.7-7-2021.170285.

Vahedifar, M.A., Akhtarshenas, A., Sabbaghian, M. and Rafatpanah, M., 2023. Information modified K-nearest neighbor. [online] Available at: http://arxiv.org/abs/2312.01991) [Accessed 31 May 2025].

Vujović, Ž., 2021. Classification model evaluation metrics. International Journal of Advanced Computer Science and Applications, 12(6), pp.599–606. doi: 10.14569/IJACSA.2021.0120670.

Wani, A.R., Rana, Q.P., Saxena, U. and Pandey, N., 2019. Analysis and detection of DDoS attacks on cloud computing environment using machine learning techniques. In: 2019 Amity International Conference on Artificial Intelligence (AICAI). pp.870–875. doi: 10.1109/AICAI.2019.8701238.

Xiao, P., Qu, W., Qi, H. and Li, Z., 2015. Detecting DDoS attacks against data center with correlation analysis. Computer Communications, 67, pp.66–74. doi: 10.1016/j.comcom.2015.06.012.

Zekri, M., El Kafhali, S., Aboutabit, N. and Saadi, Y., 2018. DDoS attack detection using machine learning techniques in cloud computing environments. In: 2017 International Conference on Cloud Computing Technology and Science (CloudTech). pp.1–7. doi: 10.1109/CloudTech.2017.8284731.

Zhang, Y., Zhao, Z. and Zheng, J., 2020. CatBoost: A new approach for estimating daily reference crop evapotranspiration in arid and semi-arid regions of Northern China. Journal of Hydrology, 588, p.125087. doi: 10.1016/j.jhydrol.2020.125087.