

Malware detection using machine learning

Gauri Dhongade *

Omprakash Chandrakar [†]

Department of Information Technology

MATS University

Raipur 492001

Chhattisgarh

India

Rajeshree Khande [§]

Department of Technology and Management

Shri Balaji Institute of Technology & Management

Sri Balaji University, Pune

Pune 411033

Maharashtra

India

Abstract

In this era of Artificial Intelligence and huge online activities, safeguarding sensitive data and protecting personal information is essential in this era. Cyber-attacks can cause the monetary loss as well as reputational loss of the victim organizations. Malware is one type of cyber-attacks; it refers to malicious software that intentionally designed to damage or exploit or harm the data and networks. Detection of malware attacks using traditional based systems will fail to detect new types of variants. This Papers aims to provide ways for malware detection using the machine learning algorithms. Malware Dataset is processed using preprocessing techniques and then machine learning algorithms like Logistic Regression, Ransom Forest, Ada boost will be used to analyse the data and provide high accuracy in classification and detection of malwares. Data Preprocessing techniques and ML algorithms will give significant results as compared to traditional ways.

Subject Classification: Primary 68T09, Secondary 68M25.

Keywords: Cybersecurity, Malware detection, Machine learning, Classification, Random forest, Artificial intelligence.

* E-mail: gauridhongade13@gmail.com (Corresponding Author)

[†] E-mail: dromprakashc@matsuniversity.ac.in

[§] E-mail: rajeshreekhande@gmail.com

Introduction

In this technical era heavy usage of internet and online activities lead to risk of vulnerabilities and cyber-attacks. From Banking, communication, education, business, e commerce, healthcare domain, almost all day-to-day activities are relying on the internet. [2] Along with global connect and convenience there is risk of cyber-attacks and frauds. There is need to monitor the increase in the vulnerability of sensitive data due to cyber-attacks, cyber-threats, cyber-crimes, and malware. [19] Securing data using traditional techniques have certain challenges because of new variants of cyber-attacks. Artificial Intelligence can provide solution to overcome these challenges. With the help of Machine Learning algorithms, Deep learning techniques and big size high quality original dataset will give more accurate results for classification and detection of malware attacks [18].

Over the past few years, the exponential growth of digital technologies has brought with it a heightened risk of cybersecurity attacks, and malware is one of the most common and devastating types of attack. [9] Malware, or malicious software, encompasses viruses, worms, trojans, ransomware, and other types of harmful programs that are meant to interfere with, harm, or obtain unauthorized access to computer systems. [20] Legacy signature-based and rule-based malware detection techniques find it difficult to cope with the changing nature of zero-day attacks and polymorphic malware, which calls for more intelligent and adaptive security technologies.

Machine Learning (ML) has proven to be a strong solution for automated malware detection and classification, allowing systems to detect threats based on patterns and behaviours instead of pre-defined signatures. [8] ML models are able to process large-scale datasets, derive useful features, and accurately classify malware, enhancing real-time threat detection. This study examines the performance of different supervised learning algorithms, including Random Forest, Support Vector Machines (SVM), Gradient Boosting, and Ada Boost models, in identifying and classifying malware.

Malware:

Malware is a computer program or software intentionally designed to damage systems, data, networks. It is used to destroy files, data, devices, networks or software systems. Various types of malware attacks are present like adware, worms, ransomware etc. [16] The basic purpose of any cyber-attack or malware attacks monetary gain.

Cyber Criminals are using modern techniques of attacks so that traditional intrusion detection system failed to identify it. [3] Phishing is a common technique used for attack. In phishing victim clicks the malicious links and get into attack. Cyber attacker can get into the corporate network via weak points or vulnerabilities in the network. After getting into the network the attacker can move around the data, steal sensitive information, access confidential data and damage system [4], [5].

For example, if we consider the case of ransomware attack, attacker attacks the system and locks system or encrypts the important files and data. Ransomware attacker sends the ransom note to victim for Ransom amount to be paid in the form of Cryptocurrency. [7] Once the ransom amount is received the attacker releases the decryption key. Ransomware attack not only financially damages the organization but also cause the reputational damage [13], [17].

Related Work:

In [25] author studied different malware detection mechanisms using AI. Author addresses the current challenges like changing nature of malwares in detection and classification of malwares. It is intended to make new researchers aware of the research process in AI-based malware detection and set the directions for future research by describing fundamental concepts, analysis techniques, and the interconnection among different studies.

One of paper discusses the shortcomings of signature-based methods for detecting unknown malware by utilizing freely available tools to monitor malware behavior. In this paper main focus is on the on the analysis of malware behavior, features, and attributes through different analytical methods to develop behavioral-based malware signatures [26].

In the paper author investigate malware analysis and detection through machine learning (ML) algorithms to address more complex polymorphic malware. The algorithmic performance of Decision Trees (DT), Convolutional Neural Networks (CNN), and Support Vector Machines (SVM) is compared, with DT performing 99% accurately and with low false positives [23].

In the paper [24], the authors recommend behavior-based detection techniques, derived from static and dynamic analysis, and how different ML techniques can be used to predict the classification of malware and its family [24].

One research paper discusses Artificial Intelligence (AI) based methods of malware detection and prevention and identifies the growing menace of malware to computer systems and their stakeholders. The research focuses on the protection of data from fraudulent transactions and investigates AI, Machine Learning, and Deep Learning algorithms for the detection of malware activities [27].

In [25] author explained challenges in malware detention using traditional ways, how AI can be used for malware detection and current scenario. Detection of evasive malwares depend on quality of malware datasets; Quality of features used for model training and tools used for analysis. This review article focusses on the use of AI tools and techniques for malware detection.

In the review paper [7] author comprehensively states the various methods used for Ransomware detection, how traditional methods are incapable in detection of zero day or new types. Author highlighted the major research challenges such as evasive techniques, quality and availability of datasets and Model building and evaluation in this domain.

Methodology: This research paper uses following methodology shown in Figure 1. To carry out this research, Malware dataset was downloaded and stored on the drive. Data Preprocessing techniques like data cleaning, processing, balancing, feature engineering techniques are applied on the data and 13 features are selected for model building. To carry out this experiment dataset is divided as 80:20. For training 80% data is used and for testing 20% data is used. After successful testing the results are analysed and compared and discussed in last section. Following diagram shows the methodology used.

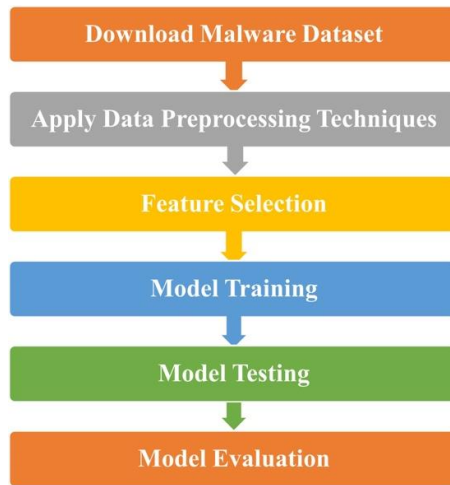


Figure 1
Proposed Methodology

Dataset Description: In this study Malware dataset is used.

Data Source: <https://www.kaggle.com/datasets/gaurirahuldhongade/malwareds>

Size: Malware Dataset contains 138048 records and 57 columns. Dataset contains 2 categorical columns, 10 float-based features and remaining 45 integer-based features.

Features: Following Table 1 gives description about features in a dataset.

Table 1
Dataset Description

Sr.No	Column Name	Description / Meaning
1	Name	Name of the executable (PE) file
2	md5	MD5 hash value uniquely identifying the file
3	Machine	Target machine architecture (e.g., x86, x64)
4	SizeOfOptionalHeader	Size of the optional header in the PE file
5	Characteristics	Flags describing file attributes and behavior
6	MajorLinkerVersion	Major version of the linker used to build the file

7	MinorLinkerVersion	Minor version of the linker
8	SizeOfCode	Total size of executable code
9	SizeOfInitializedData	Size of initialized data sections
10	SizeOfUninitializedData	Size of uninitialized (BSS) data
11	AddressOfEntryPoint	Memory address where execution begins
12	BaseOfCode	Base address of the code section
13	BaseOfData	Base address of the data section
14	ImageBase	Preferred load address in memory
15	SectionAlignment	Alignment of sections when loaded into memory
16	FileAlignment	Alignment of sections in the file on disk
17	MajorOperatingSystemVersion	Required major OS version
18	MinorOperatingSystemVersion	Required minor OS version
19	MajorImageVersion	Major version of the image
20	MinorImageVersion	Minor version of the image
21	MajorSubsystemVersion	Major version of the subsystem
22	MinorSubsystemVersion	Minor version of the subsystem
23	SizeOfImage	Total size of the image in memory
24	SizeOfHeaders	Size of all headers combined
25	Checksum	Checksum value of the PE file
26	Subsystem	Subsystem required to run the file (GUI, Console, etc.)
27	DllCharacteristics	DLL-related characteristics and security flags
28	SizeOfStackReserve	Memory reserved for stack
29	SizeOfStackCommit	Memory committed for stack
30	SizeOfHeapReserve	Memory reserved for heap
31	SizeOfHeapCommit	Memory committed for heap
32	LoaderFlags	Loader-specific flags
33	NumberOfRvaAndSizes	Number of data directory entries
34	SectionsNb	Total number of sections in the PE file
35	SectionsMeanEntropy	Average entropy of sections (packing indicator)
36	SectionsMinEntropy	Minimum section entropy
37	SectionsMaxEntropy	Maximum section entropy
38	SectionsMeanRawsize	Mean raw size of sections
39	SectionsMinRawsize	Minimum raw size of sections
40	SectionsMaxRawsize	Maximum raw size of sections
41	SectionsMeanVirtualsize	Mean virtual size of sections
42	SectionsMinVirtualsize	Minimum virtual size of sections
43	SectionsMaxVirtualsize	Maximum virtual size of sections
44	ImportsNbDLL	Number of imported DLLs
45	ImportsNb	Total number of imported API functions
46	ImportsNbOrdinal	Number of imports using ordinal values
47	ExportNb	Number of exported functions
48	ResourcesNb	Number of embedded resources
49	ResourcesMeanEntropy	Mean entropy of resources
50	ResourcesMinEntropy	Minimum resource entropy
51	ResourcesMaxEntropy	Maximum resource entropy
52	ResourcesMeanSize	Mean size of resources

53	ResourcesMinSize	Minimum resource size
54	ResourcesMaxSize	Maximum resource size
55	LoadConfigurationSize	Size of load configuration structure
56	VersionInformationSize	Size of version information metadata
57	legitimate	Class label (1 = Legitimate, 0 = Malware/Ransomware)

II. Data preprocessing – Data

Preprocessing is important in malware detection as dataset may contains noisy, inconsistent data, high dimensional data, imbalanced data. [6] Preprocessing involves clearing of missing values, noisy, redundant data. [22] Raw data needs to be pre-processed for further use in machine learning for model training and testing. Preprocessing methods, Feature scaling and normalization methods helps in increasing the accuracy of model [21].

III. Feature Selection

This dataset contains 57 features, all features will not take part in the detection of malware, so there is need of important feature selection. Feature selection techniques are used to select important features from the dataset for implementation of Machine learning algorithms. Here in this study Extra Tree Classifier is used for feature selection.

IV. Experimental Work and Results

In our research we used supervised machine learning algorithms and classifiers for detection of malware. Here Logistic regression, KNN, Random Forest, AdaBoost, Gradient Boost algorithms are trained and tested for malware detection. [12], [14] Theoretical details about each algorithm is given in this section.

Logistic regression

Logistic regression is a simple method used to predict outcomes that have two possible results, like yes/no or true/false. It looks at the relationship between input factors (like age or income) and the chance of a certain outcome happening. It uses a special S-shaped curve called the sigmoid function to turn these inputs into a probability between 0 and 1. If the result is above 0.5, we usually predict "yes"; otherwise, "no". The model learns the best values for its parameters using a method called maximum likelihood estimation, and its performance can be measured using accuracy, F1 score, and similar metrics [1].

Mathematics behind Algorithm

Hypothesis Function

The logistic regression model predicts probability using the **sigmoid (logistic) function**:

$$h(\theta(x)) = 1 / [1 + e^{-(\theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \dots + \theta_n x_n)}]$$

where:

$h\theta(x)$: predicted probability that $y=1$

$\theta_1+\theta_2+\dots+\theta_n$: model parameters

$x_1, x_2, x_3, \dots, x_n$: input features

This is also called the **sigmoid function** — it converts any real value of z into a probability between **0 and 1**.

KNN

k-Nearest Neighbours Classification (k-NN) is one of the most straightforward methods for classification and regression wherein predictions are made from the k closest data points ascertained using a distance metric, such as Euclidean distance. In classification, the function assigns a label to the case, which dominates within the neighbourhood, while the case averages for regression. The performance of the model is affected by the choice of k : small k increases sensitivity to noise (overfitting), while large k can smooth out important patterns. k-NN is particularly effective when the neighbouring data points yield similar results, however, in larger datasets where distance needs to be calculated for every prediction, it becomes slower.

Mathematics behind Algorithm

Distance Formula (Euclidean Distance): Used to measure closeness between two data points X_i and X_j :

$$d(X_i, X_j) = \sqrt{\sum_{k=1}^n (X_{ik} - X_{jk})^2}$$

where

- n : number of features
- X_{ik}, X_{jk} : values of the K th feature of points i and j

Other Distance Metrics (optional)

- **Manhattan Distance:**

$$d(X_i, X_j) = \sum_{k=1}^n |X_{ik} - X_{jk}|$$

- **Minkowski Distance (generalized form):**

$$d(X_i, X_j) = \left(\sum_{k=1}^n |X_{ik} - X_{jk}|^p \right)^{1/p}$$

(where $p=2$ gives Euclidean, $p=1$ gives Manhattan)

Prediction Rule

After finding the **K nearest neighbours**, prediction is made as:

- **For Classification:**

$$y_{\Lambda} = \text{mode}(y_1, y_2, \dots, y_k)$$

(the majority class among K neighbours)

- **For Regression:**

$$y = \frac{1}{k} \sum_{i=1}^k y_i$$

(average of K nearest neighbour values)

Random Forest

It is one of the algorithms of Machine Learning used for the classification of tabular data. In this algorithm many trees are created using random parts of data and features. Every tree in this algorithm gives its own prediction and the final result is based on the majority vote from all predictions of trees. This technique of result prediction using majority voting helps reduce mistakes and prevents overfitting, which is a common problem with a single decision tree. Random Forest works well for both classification and regression, and it gives better accuracy by combining the results of multiple trees—like asking a group of experts and going with the most common answer.

Mathematics of Algorithm

Individual Tree Prediction

Each decision tree $h_b(x)$ in the forest gives its own class prediction:

$$h_b = \text{DecisionTree}_b(X) \quad b = 1, 2, \dots, B$$

where

- B: total number of trees
- X: input feature vector

Final Random Forest Prediction (Classification)

The **majority voting rule** is used:

$$y = \text{mode}\{h_1(x), h_2(x), \dots, h_B(x)\}$$

Final Prediction is based on the majority votes received from all trees for a particular class.

For Probability Estimation

The probability of class c is computed as:

$$P(y = c | x) = \frac{1}{B} \sum_{b=1}^B 1(h_b(x) = c)$$

where

- $1(\cdot)$ = indicator function (1 if true, 0 if false)

For Regression (if used)

$$y = \frac{1}{B} \sum_{b=1}^B h_b(x)$$

Gradient Boost

It is one of the best ensemble techniques used in Machine learning for the prediction. It builds the model by adding weak models like decision trees.

Gradient decent and Loss function is used in this classifier. In this model final prediction is based on outcome of all models.

Model Representation

Gradient Boosting builds an **additive model** of weak learners (usually decision trees):

$$F_M(X) = \sum_{m=1}^M v_m h_m(x)$$

where

- $F_M(X)$: final model after M iterations
- $h_m(x)$: weak learner (tree) added at step m
- v_m : learning rate or step size

Final Prediction

For **classification**:

$$P(y = 1|x) = 1 / (1 + e^{(-FM(x))})$$

and the final predicted class:

$$y^{\wedge} = \begin{cases} 1, & \text{if } P(y = 1|x) \geq 0.5 \\ 0, & \text{if } P(y = 1|x) < 0.5 \end{cases}$$

Ada Boost

Ada Boost the AdaBoost (Adaptive Boosting) Classifier is an ensemble learning method that uses several weak classifiers (typically decision trees) to build a strong classifier. It gives more weight to misclassified instances, compelling the subsequent model to concentrate on more difficult cases [15].

Model Representation

AdaBoost combines multiple **weak learners** (usually decision stumps) into a **strong classifier**:

$$F(x) = \sum_{m=1}^M a_m h_m(x)$$

where

- $h_m(x)$: weak learner at iteration m
- a_m : weight (importance) of that learner
- M: total number of weak learners

Final Strong Classifier

Combine all weak learners into a final prediction:

$$H(x) = \text{sign} \left(\sum_{m=1}^M a_m h_m(x) \right)$$

Training and Testing of Model

In this experiment, dataset is divided as 80:20. 80% data is used for training the model and 20% data is used for testing of model. The model is trained using given 80% dataset and algorithms Logistic Regression, KNN,

ransom Forest, Gradient Boost, Ada Boost classifiers. After successful model training, it is tested on test dataset and compared for results.

Performance Metrics

Machine learning performance metrics are quantitative measures used to evaluate the effectiveness and reliability of models, ensuring they meet desired objectives. Classification Metrics *Accuracy*, *Precision*, *Recall*, *F1 Score* are used shown in Table 2 and Figure 2, [10], [11].

Table 2
Result Analysis

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	70.00%	0	0	0
KNN	98.98%	98%	99%	98%
Random Forest	99.36%	99%	99%	99%
Gradient Boost	98.70%	99%	98%	98%
Ada Boost	98.33%	98%	97%	97%

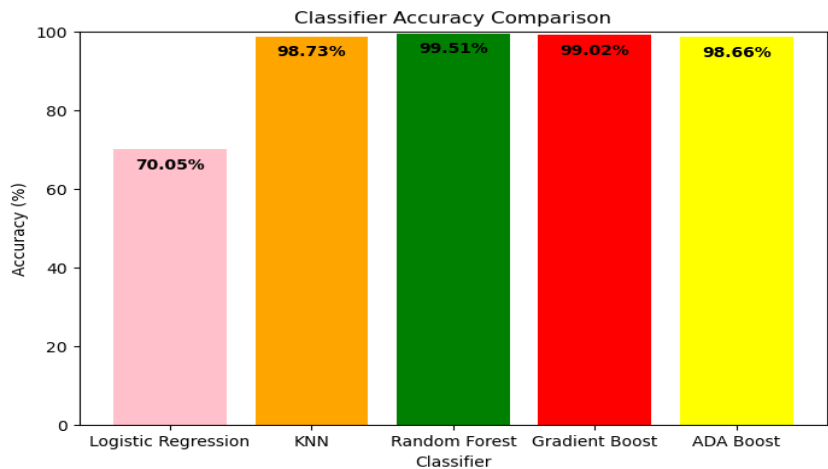


Figure 2
Comparison of Classification Accuracy Across Different Machine Learning Models

Conclusion

This study proves the efficacy of machine learning models in malware detection and classification. The findings show that ensemble learning techniques like Random Forest, Gradient Boosting, and AdaBoost outperform conventional models like Logistic Regression significantly.

Major findings from the comparison:

- Random Forest recorded the highest accuracy (99.36%) with high precision, recall, and F1-score (99% each), and hence it is the best classifier for malware detection.
- K-Nearest Neighbours' (KNN) also had good performance with 98.98% accuracy, with high recall (99%), and is thus good at reducing false negatives.
- Gradient Boosting and AdaBoost had 98.70% and 98.33% accuracy, respectively, with strong but slightly weaker performance compared to Random Forest.
- Logistic Regression completely failed with low precision, recall, and F1-score, which means that it is not good for this kind of malware classification problem.

These findings emphasize the significance of ensemble learning methods in cybersecurity use. Random Forest is the most appropriate model because it has better accuracy, stability, and balanced performance.

Future Scope:

There is need of real time malware detection system. Future research can investigate deep learning methods, feature selection methods, and real-time malware detection models to further improve detection efficiency and accuracy.

References

- [1] M. S. Akhtar and T. Feng, "Evaluation of machine learning algorithms for malware detection," *Sensors*, vol. 23, no. 2, Art. no. 946 (2023), doi: 10.3390/s23020946.
- [2] M. S. Akhtar and T. Feng, "Malware analysis and detection using machine learning algorithms," *Symmetry*, vol. 14, no. 11, Art. no. 2304 (2022), doi: 10.3390/sym14112304.
- [3] E. J. Alqahtani, R. Zagrouba, and A. Almuhaideb, "A survey on Android malware detection techniques using machine learning algorithms," in *Proc. 2019 6th Int. Conf. Software Defined Systems (SDS)*, Rome, Italy, pp. 110–117 (2019), doi: 10.1109/SDS.2019.8768729.
- [4] Ö. A. Aslan and R. Samet, "A comprehensive review on malware detection approaches," *IEEE Access*, vol. 8, pp. 6249–6271 (2020), doi: 10.1109/ACCESS.2019.2963724.
- [5] M. Azeem, D. Khan, S. Iftikhar, S. Bawazeer, and M. Alzahrani, "Analyzing and comparing the effectiveness of malware detection:

- A study of machine learning approaches," *Heliyon*, vol. 10, Art. no. e23574 (2024), doi: 10.1016/j.heliyon.2023.e23574.
- [6] C. Fan, M. Chen, X. Wang, J. Wang, and B. Huang, "A review on data preprocessing techniques toward efficient and reliable knowledge discovery from building operational data," *Front. Energy Res.*, vol. 9, Art. no. 652801 (2021), doi: 10.3389/fenrg.2021.652801.
 - [7] J. Ferdous, R. Islam, A. Mahboubi, and M. Z. Islam, "AI-based ransomware detection: A comprehensive review," *IEEE Access*, vol. 12, pp. 136666–136695 (2024), doi: 10.1109/ACCESS.2024.3461965.
 - [8] D. Gavriluț, M. Cimpoeșu, D. Anton, and L. Ciortuz, "Malware detection using machine learning," in *Proc. Int. Multiconf. Comput. Sci. Inf. Technol.*, pp. 735–741 (2009).
 - [9] S. Gupta, "Types of malware and its analysis," *Int. J. Sci. Eng. Res.*, vol. 4, no. 1, pp. 1–13 (2013).
 - [10] M. J. H. Faruk, H. Shahriar, M. Valero, F. L. Barsha, S. Sobhan, M. A. Khan, M. Whitman, A. Cuzzocrea, D. Lo, A. Rahman, and F. Wu, "Malware detection and prevention using artificial intelligence techniques," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Orlando, FL, USA, pp. 5369–5377 (2021), doi: 10.1109/BigData52589.2021.9671434.
 - [11] U. Inayat, M. F. Zia, F. Ali, S. M. Ali, H. M. A. Khan, and W. Noor, "Comprehensive review of malware detection techniques," in *Proc. Int. Conf. Innovative Comput. (ICIC)*, pp. 1–6 (2021).
 - [12] J. Landage and M. P. Wankhade, "Malware and malware detection techniques: A survey," *Int. J. Eng. Res.*, vol. 2, no. 12, pp. 61–68 (2013).
 - [13] S. J. Lee, H. Y. Shim, Y. R. Lee, T. R. Park, S. H. Park, and I. G. Lee, "Study on systematic ransomware detection techniques," in *Proc. Int. Conf. Adv. Commun. Technol. (ICACT)*, pp. 297–301 (2022).
 - [14] M. Naseer, J. F. Rusdi, N. M. Shanono, S. Salam, Z. B. Muslim, N. A. Abu, and I. Abadi, "Malware detection: Issues and challenges," *J. Phys. Conf. Ser.*, vol. 1807, no. 1, Art. no. 012011 (2021).
 - [15] J. Pavithra and S. Selvakumara Samy, "A comparative study on detection of malware and benign on the internet using machine learning classifiers," *Math. Probl. Eng.*, vol. 2022, Art. no. 4893390 (2022).
 - [16] N. Selamat and F. Ali, "Comparison of malware detection techniques using machine learning algorithm," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 16, no. 1, pp. 435–440 (2019).
 - [17] G. Dhongade, O. Chandrakar, and R. Khande, "Evolution of ransomware attacks: A basic review," *Int. J. Sci. Res. Sci. Eng. Technol.*, vol. 12, no. 4, pp. 339–346 (2025).

- [18] G. Dhongade, D. Chandrakar, and R. Khande, "Enhancing cyber security: A study of data preprocessing techniques for cyber security datasets," *Int. J. Sci. Res. Sci. Technol.*, vol. 11, no. 5, pp. 71–75 (2024).
- [19] A. Bensaoud, J. Kalita, and M. Bensaoud, "A survey of malware detection using deep learning," *Mach. Learn. Appl.*, vol. 16, Art. no. 100546 (2024), doi: 10.1016/j.mlwa.2024.100546.
- [20] K. R. Prasad, R. R. Kumar, S. Suneel, and R. Nuvvusetty, "Exploring diverse network threats on cyber-physical systems," *J. Discrete Math. Sci. Cryptogr.*, vol. 28, no. 6, pp. 2561–2571 (2025), doi: 10.47974/JD-MS-2340.
- [21] A. P. Singh and M. Singh, "MS real-time malware detection in encrypted network traffic using machine learning with time-based features," *J. Discrete Math. Sci. Cryptogr.*, vol. 26, pp. 841–850 (2023).
- [22] A. K. Sharma, N. Babbar, P. Pal, and N. K. Verma, "AI-powered framework for real-time dark web monitoring and cybersecurity threat detection," *J. Discrete Math. Sci. Cryptogr.*, vol. 28, no. 8, pp. 2955–2966 (2025), doi: 10.47974/JDMS-2440.
- [23] M. S. Akhtar and T. Feng, "Malware analysis and detection using machine learning algorithms," *Symmetry*, vol. 14, no. 11, Art. no. 2304 (2022), doi: 10.3390/sym14112304.
- [24] S. Agarkar and S. Ghosh, "Malware detection & classification using machine learning," in *Proc. 2020 IEEE Int. Symp. Sustainable Energy, Signal Process. Cyber Security (iSSSC)*, Gunupur, Odisha, India, pp. 1–6 (2020), doi: 10.1109/iSSSC50941.2020.9358835.
- [25] M. G. Gaber, M. Ahmed, and H. Janicke, "Malware detection with artificial intelligence: A systematic literature review," *ACM Comput. Surv.*, vol. 56, no. 6, Art. no. 148, pp. 1–33 (2024), doi: 10.1145/3638552.
- [26] O. P. Samantray, S. N. Tripathy, and S. K. Das, "Notice of violation of IEEE publication principles: A study to understand malware behavior through malware analysis," in *Proc. 2019 IEEE Int. Conf. System, Comput., Autom. Networking (ICSCAN)*, Pondicherry, India, pp. 1–5 (2019), doi: 10.1109/ICSCAN.2019.8878680.
- [27] M. J. H. Faruk, H. Shahriar, M. Valero, F. L. Barsha, S. Sobhan, M. A. Khan, M. Whitman, A. Cuzzocrea, D. Lo, A. Rahman, and F. Wu, "Malware detection and prevention using artificial intelligence techniques," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Orlando, FL, USA, pp. 5369–5377 (2021), doi: 10.1109/BigData52589.2021.9671434.