IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 8, April 2025
Impact Factor: 7.67

Malware Detector using Machine Learning

Harsh Bhardwaj, Maniya Tadhiyal, Lakshay Kamboj
Dronacharya College of Engineering, Gurgaon, India

Abstract: *Malware is a huge cybersecurity problem due to the exponential growth of digital technologies. It causes global financial and data losses. Traditional signature-based and heuristic detection methods fail to detect new and complicated malware strains. In recent years, machine learning (ML) has become a strong alternative, identifying both known and unknown malware by learning patterns from static and dynamic properties. This paper examines ML malware detection using past research and data. It covers supervised, unsupervised, and deep learning models, their assessment criteria, and practical applications. The article addresses dataset imbalance, generalisation, and explainability as well as future prospects including hybrid modelling and privacy-preserving techniques. This secondary research stresses the potential of ML to transform malware detection systems and the need for continued progress to combat sophisticated cyber attacks.*

Keywords: malware detection, machine learning, cybersecurity, deep learning, malware datasets, classification algorithms, anomaly detection, real-time detection

I. INTRODUCTION

Malware, one of the most pervasive and dangerous problems in today's digital society, has increased crimes. Malware is software designed to harm computers, servers, customers, or networks. It includes viruses, Trojan horses,worms, spyware, ransomware, and rootkits [1]. Malware can ruin infrastructure, steal data, hijack resources, and impair system performance. With more people, governments, and organisations using digital technology, malware's impact on cybersecurity has expanded. Malware can affect cybersecurity, causing financial losses, reputational damage, and national security dangers. Malware detection systems are now crucial to cybersecurity designs due to these growing threats. Rule-based and signature-based methods are still employed, although they may miss new or disguised malware, especially polymorphic or metamorphic malware. ML based malware detection has grown in popularity due to its ability to learn from data, discover complex patterns, and detect previously unknown malware with improved accuracy and flexibility.

MLis essential to malware detection because it lets computers learn from vast quantities of data and adapt to new threats without manual retraining [2]. ML techniques use supervised, unsupervised, and deep learning models to analyse system call records, binary features, network traffic, and behavioural patterns to distinguish legitimate from malicious activity. ML-based detectors outperform static or manual methods in scalability, reaction time, and zero-day attack detection.

This study examines machine learning-based malware detection using secondary research. The plan includes reviewing current literature, assessing popular ML models, comparing their performance, and identifying challenges and future research directions. This study combines previous studies to understand how MLis changing malware detection.

General Architecture of ML-Based Malware Detection System

DOI: 10.48175/IJARSCT-25518

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 8, April 2025
Impact Factor: 7.67

II. BACKGROUND AND LITERATURE REVIEW

2.1 Traditional Malware Detection Methods

Malware detection has long been part of cybersecurity and employed many approaches before intelligent systems. Signature-based detection, which searches for harmful code "signatures" or patterns, is the gold standard [3]. This method works effectively against recognised threats, but new or polymorphic malware can disguise its code. Heuristic-based malware detection looks for unexpected code or instructions. Despite its high false-positive rate, it can detect unknown malware variants. Behavior-based detection watches for suspect application activity during runtime, such as unauthorised file changes or unusual network traffic. This method detects and blocks encrypted or obfuscated malware better, but it is resource-intensive and evasion-prone.

2.2 Limitations of Traditional Techniques

Cybersecurity has used classic detection methods for decades, but they have severe limitations. Because signature-based systems only employ their databases for security, they are vulnerable to zero-day assaults and disguised malware. Heuristic methods may generate too many false positives in business [4]. Behavior-based methods are promising but resource-intensive and can't manage modern attackers' speed and volume. Conventional systems typically lack adaptability and fail to generalise to new malware. As these vulnerabilities have been uncovered, MLhas enabled more intelligent, automated, and scalable malware detection methods.

2.3 Emergence and Evolution of MLin Malware Detection

Due to the limitations of prior malware detection methods, MLhas become a popular technology. ML techniques can automatically learn patterns from enormous datasets, adapt to new hazards, and operate in real time. MLmodels, unlike static signature databases, may discover new viruses by examining code and behaviour. ML in this field has progressed from Decision Trees and Support Vector Machines (SVM) to ensemble techniques and deep learning architectures. Enhanced feature extraction methods and more labelled malware datasets have increased this field's research.

2.4 Key Insights from Literature

A analysis of recent papers found that MLalgorithms are widely employed for malware detection. Popular supervised learning algorithms include SVM, Random Forests, k-NN, and Naïve Bayes [5]. These models classify malware well by studying API calls, opcode sequences, and permissions. Convolutional Neural Networks (CNNs) for malware classification based on images and LSTM models and RNNs for pattern recognition in sequential data like file operations or network logs have been successful in deep learning. According to studies, high-quality datasets are needed to train and evaluate these models. Public datasets like VirusShare, EMBER, CICMalDroid, and Malimg help make research repeatable.

These datasets usually contain benign and hazardous samples and may contain raw binaries, extracted features, or dynamic behaviour logs. Opcode frequency, API calls, byte sequences, and system calls are retrieved in feature engineering and determine ML model efficacy.

Studies use F1-score, recall, accuracy, precision, and Area Under the ROC Curve to evaluate performance. Most models exceeded 90% accuracy when trained on carefully selected datasets. Some papers say overfitting and testing on imbalanced datasets are instances of how high accuracy does not guarantee robustness. Research continues to generalise across malware types while minimising false positives.

III. METHODOLOGY

This paper evaluates and analyses machine learning-based malware detection studies using secondary sources. Primary data sources include VirusShare, VirusTotal, and CICMalDroid malware databases, whitepapers, technical studies, surveys, and peer-reviewed research articles. This study uses literature from the recent five to 10 years, making it current. Topic relevance, methodology clarity, performance metrics availability, and MLvirus detection were considered for selecting studies. Empirical investigations of ML models utilising standard datasets were

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 8, April 2025
Impact Factor: 7.67

favoured. The review comprised feature extraction, model evaluation, and practical implementation papers. This method explains machine learning's involvement in malware detection and uses verified secondary data.

IV. TYPES OF ML MODELS USED

4.1 Supervised Learning

The simplicity of supervised learning's mechanism for mapping input information to known output labels makes it a popular choice among researchers working on virus identification. These models are taught using datasets that have been labelled as malicious or benign. Here we may find some of the most popular algorithms, such as decision trees,random forests, and support vector machines (SVMs).

It is common practice to employ SVMs for malware classification using extracted features like opcode frequencies or API calls because of their great accuracy and capacity to handle high-dimensional data [6]. Random forests, which are ensembles of decision trees, enhance generalisation and decrease the danger of overfitting. Decision trees, on the one hand, offer a straightforward and efficient framework for rule-based categorisation. When there is an abundance of high-quality labelled data, these models excel at learning the unique patterns of known malware. New or obfuscated malware that doesn't appear in the training data can be difficult for them to identify.

4.2 Unsupervised Learning

Unsupervised learning offers an alternative approach that is particularly useful in identifying previously unknown or evolving malware. Unlike supervised models, unsupervised methods operate without labeled outputs, instead discovering hidden structures or anomalies within the data. Clustering techniques, such as k-means and DBSCAN, group similar samples together based on feature similarities, which can highlight unusual patterns that may indicate malicious behavior [7]. Anomaly detection methods also fall under this category and are designed to detect outliers that significantly deviate from normal system behavior. These techniques are especially valuable for zero-day attacks, where new malware variants may not yet be labeled or documented. Although unsupervised models can provide insights into unknown threats, they often require fine-tuning and expert interpretation, and they are generally less precise than supervised models when it comes to clearly labeling threats.

4.3 Deep Learning Approaches

The use of deep learning for malware detection has grown in popularity due to the availability of large-scale datasets and the increase in processing capacity. Without human intervention, deep learning models may automatically derive high-level features from input data. Convolutional neural networks (CNNs) are a popular deep learning technique that have shown promise in image-based malware detection. This method involves greyscale image conversion of malware binaries and CNN training for visual structure-based classification [8]. Especially when it comes to detecting polymorphic malware, which changes its code structure but keeps its essential operation the same, this strategy has shown to be very accurate and resilient. The recurrent neural network (RNN) and its enhanced version, long short-term memory (LSTM), are another important architecture in deep learning. System calls, network traffic, or API sequences are examples of time-series data that these models excel at analysing for sequence-based malware identification. For real-time detection systems and dynamic analysis, their capacity to record temporal interdependence is perfect.

4.4 Comparative Effectiveness of Models

Literature comparing these different MLmodels indicates varying degrees of effectiveness depending on the application context, dataset quality, and feature selection. Supervised models like SVM and random forests consistently report high performance on static datasets, with accuracy often exceeding 90%. However, their effectiveness can decline when faced with novel malware or imbalanced data. Unsupervised models, while useful in exploratory analysis and unknown threat detection, are generally less accurate and more prone to false positives [9]. When it comes to collecting complicated patterns and managing raw input data without considerable preprocessing, deep learning models, especially CNNs and RNNs, have proven to be the best.However, they require large amounts of labeled data and substantial computational resources. Overall, hybrid approaches that combine multiple techniques, such as integrating

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 8, April 2025
Impact Factor: 7.67

supervised learning with anomaly detection or combining static and dynamic analysis features, are increasingly being explored for their ability to enhance detection accuracy and adaptability.

V. DATASET CHARACTERISTICS AND FEATURE ENGINEERING

5.1 Commonly Used Datasets

To make machine learning methods better at finding malware, it's important to have access to a wide range of high-quality samples. There are a number of open-source statistics that have become standards in the field.

The CICMalDroid dataset, which was made to find malware on Android, has a good mix of static and dynamic traits from real-life apps [10]. Another dataset that is used a lot for training and testing static malware algorithms is EMBER. There are many traits in it that were taken from executable files, which makes it good for training strong models. Malimg turns malware binaries into greyscale pictures that can be analysed visually for patterns. This is called image-based malware categorisation. VirusShare is a huge collection of malware samples that researchers have shared, but they often need to be processed and labelled in a lot of detail before they can be used in machine learning apps.

5.2 Key Features in Malware Detection

For machine learning models to work well, they need to be able to select the right traits. we can see parts of malware that aren't working without actually running it. At the code level, these include things like API calls, rights asked for, opcode sequences, header data, and file metadata. They make things safer to use and easier to get, but they might not work on malware that is hidden or protected [11]. Things that are recorded while a program is running are called dynamic features. Some of these are system calls, file changes, memory procedures, network activity, and CPU use. Dynamic analysis is better at finding advanced malware that hides dangerous behaviour in static code, but it takes more time and resources, and sandbox detection methods can get around it.

5.3 Feature Selection Techniques

Because malware datasets have a lot of dimensions, feature selection is an important part of making models that work well. Many people use mutual information, recursive feature removal, information gain, and chi-square tests to find the most useful features and get rid of the ones that aren't needed. By getting rid of unnecessary or noisy data, these methods help make the model simpler, training go faster, and generalisation better. For picking out important traits in some studies, methods like L1 regularisation in logistic regression or decision tree-based feature importance rankings are also used.

5.4 Data Preprocessing Techniques

Before feeding data into machine learning models, preprocessing is necessary to standardize and prepare the input. For numerical features, scaling methods such as min-max normalization or standardization are applied to ensure consistent ranges across attributes [12]. Categorical data, such as permissions or file types, are typically encoded using techniques like one-hot encoding or label encoding. Additionally, handling class imbalance through methods such as SMOTE (Synthetic Minority Over-sampling Technique) or undersampling helps in preventing biased learning towards the dominant class. Feature vectors are often cleaned to remove missing or corrupted entries, and in the case of text-based features like API sequences, tokenization and vectorization methods such as TF-IDF or word embeddings may be applied. These preprocessing steps play a critical role in ensuring that the MLmodel receives clean, structured, and informative data for accurate malware detection.

VI. EVALUATION METRICS AND RESULTS FROM LITERATURE

6.1 Common Evaluation Metrics

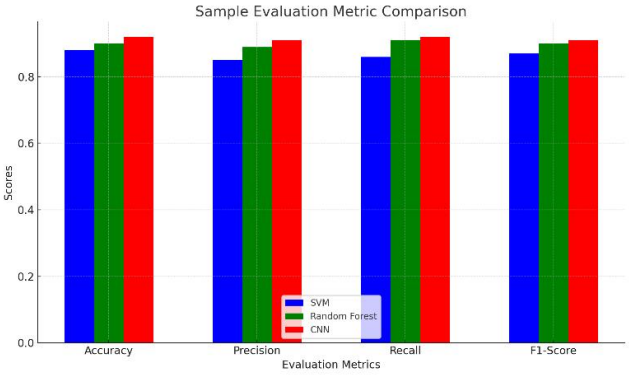
In the context of MLfor malware detection, several performance indicators are employed to assess the efficacy of the models. An often-cited indicator, accuracy measures the percentage of samples that were properly identified relative to the total. In unbalanced datasets, where harmless samples may greatly outnumber harmful ones, accuracy on its alone might be deceiving. That is why F1-score, recall, and precision all give more complex information. While recall is

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 8, April 2025
Impact Factor: 7.67

concerned with the model's capacity to identify all instances of malware, precision is concerned with the fraction of samples that are actually categorised as malware. The F1-score provides a fair assessment of the model's efficacy since it is the harmonic mean of recall and precision. The area under the receiver operating characteristic (ROC) curve (AUC) is another important statistic; it allows one to compare models independent of threshold settings and highlights the trade-off between true positive and false positive rates.

6.2 High-Performing Models from Literature

A number of high-performing models have been reported across various studies. For instance, random forest and support vector machines often achieve accuracy levels above 90% when applied to well-labeled static datasets like EMBER [13]. In deep learning approaches, CNNs used for image-based malware classification on the Malimg dataset have shown precision and recall values exceeding 95%, with F1-scores close to 0.96. RNNs and LSTMs, when applied to dynamic behavioral data, also report high recall rates, indicating their strength in capturing temporal patterns. Several hybrid models that integrate static and dynamic features or combine supervised learning with anomaly detection have further improved performance, showing robust results across different test conditions.



6.3 Trade-offs and Real-World Implications

Even though controlled experiments showed good results, using ML models to find malware in the real world comes with important trade-offs. Getting the right amount of false positives and false rejections is a big problem. A model with a high recall may catch most malware, but it could also set off a lot of false alarms, sending system admins too many files that aren't threats. On the other hand, if try to get rid of too many false positives, might miss real malware, which is a major security risk. This time, have to choose between speed and accuracy. Deep learning models might be better at finding things, but they take a lot of time and computing power, so they might not work well in real-time systems. Decision trees and Naive Bayes are two lightweight models that can make predictions faster, but they may not be as accurate.

6.4 Deployment Challenges

Putting academic models to use in real life is not easy for many reasons. Models that were trained on carefully chosen datasets might not work well in real life, where malware is always changing and using obfuscation tactics. Besides that, problems like data drift, adversarial attacks, and the need for regular retraining make adoption even harder. Integration with existing security infrastructure, following privacy laws, and being able to grow across big networks are also very important things to think about. So, even though research shows promising outcomes, real-life use needs careful tuning, constant monitoring, and methods for adaptive learning to keep working well over time.

VII. CHALLENGES AND LIMITATIONS

Despite machine learning breakthroughs in malware detection, various obstacles still prevent its widespread use. Malware developers utilise polymorphism and code obfuscation to avoid signature- or behavior-based detection, which

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 8, April 2025
Impact Factor: 7.67

is a serious challenge. When innocuous files outweigh malware samples, dataset imbalance biases models, making them less effective at recognising threats. Many algorithms fail to detect fresh or zero-day malware with new patterns due to generalisation issues. While accurate, deep learning models are frequently black boxes with minimal explainability, posing difficulties in situations that require transparency and confidence. Models that analyse sensitive data may misuse or accidentally expose user data, raising ethical and privacy concerns.These issues require a balanced approach of technical, ethical, and policy answers.

VIII. FUTURE SCOPE AND RECOMMENDATIONS

Machine learning-based malware detection systems have various intriguing future directions. Hybrid models that integrate static and dynamic analysis or supervised and unsupervised learning can increase evasion and new threat resilience. To reduce response times and virus harm, real-time detection must be improved. Federated learning allows several organisations to collaborate on model training without exchanging raw data, protecting user privacy.

Cross-platform malware is becoming more common, highlighting the need for universal models that can analyse malware for Windows, Android, and macOS.Finally, the standardization and open sharing of well-labeled and diverse datasets should be prioritized to ensure comparability of research outcomes and encourage broader collaboration within the cybersecurity research community.

IX. CONCLUSION

In conclusion, MLhas emerged as a powerful and adaptable approach for detecting and mitigating malware threats, outperforming many traditional techniques in accuracy and adaptability. This paper highlighted various MLmodels, datasets, feature engineering techniques, and evaluation metrics used in the domain of malware detection, along with the practical challenges faced during deployment. While high-performing models like random forests, CNNs, and RNNs have shown encouraging results, the dynamic and evasive nature of malware, coupled with concerns over interpretability and data privacy, present ongoing challenges. Continued research focusing on hybrid models, real-time detection, and collaborative data frameworks will be key to advancing the field. Ultimately, the application of MLin malware detection holds great promise, but its success depends on overcoming technical limitations and ensuring ethical, secure, and scalable deployment in real-world systems.

REFERENCES

[1] H. Rathore, S. Agarwal, S. K. Sahay, and M. Sewak, “Malware detection using machine learning and deep learning,” in *Proc. Int. Conf. Big Data Analytics*, Cham, Switzerland, Nov. 2018, pp. 402–411.

[2] L. Liu, B. S. Wang, B. Yu, and Q. X. Zhong, “Automatic malware classification and new malware detection using machine learning,” *Front. Inf. Technol. Electron. Eng.*, vol. 18, no. 9, pp. 1336–1347, Sep. 2017.

[3] A. Mahindru and A. L. Sangal, “MLDroid—framework for Android malware detection using machine learning techniques,” *Neural Comput. Appl.*, vol. 33, no. 10, pp. 5183–5240, May 2021.

[4] J. Senanayake, H. Kalutarage, and M. O. Al-Kadri, “Android mobile malware detection using machine learning: A systematic review,” *Electronics*, vol. 10, no. 13, p. 1606, Jul. 2021.

[5] A. Bensaoud, J. Kalita, and M. Bensaoud, “A survey of malware detection using deep learning,” *Mach. Learn. with Appl.*, vol. 16, p. 100546, Jun. 2024.

[6] S. I. Bae, G. B. Lee, and E. G. Im, “Ransomware detection using machine learning algorithms,” *Concurrency Comput. Pract. Exp.*, vol. 32, no. 18, Art. no. e5422, Sep. 2020.

[7] A. A. Majid, A. J. Alshaibi, E. Kostyuchenko, and A. Shelupanov, “A review of artificial intelligence based malware detection using deep learning,” *Mater. Today Proc.*, vol. 80, pp. 2678–2683, Jan. 2023.

[8] M. S. Akhtar and T. Feng, “Malware analysis and detection using machine learning algorithms,” *Symmetry*, vol. 14, no. 11, p. 2304, Nov. 2022.

[9] E. B. Karbab, M. Debbabi, A. Derhab, and D. Mouheb, “MalDozer: Automatic framework for Android malware detection using deep learning,” *Digit. Invest.*, vol. 24, pp. S48–S59, Mar. 2018.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 8, April 2025
Impact Factor: 7.67

[10] Z. Xu, S. Ray, P. Subramanyan, and S. Malik, "Malware detection using machine learning based analysis of virtual memory access patterns," in *Proc. Design, Automation Test Eur. Conf. Exhib. (DATE)*, Mar. 2017, pp. 169–174.
[11] O. N. Elayan and A. M. Mustafa, "Android malware detection using deep learning," *Procedia Comput. Sci.*, vol. 184, pp. 847–852, Jan. 2021.
[12] A. Kumar, K. Abhishek, K. Shah, D. Patel, Y. Jain, H. Chheda, and P. Nerurkar, "Malware detection using machine learning," in *Knowledge Graphs and Semantic Web: 2nd Iberoamerican Conf. and 1st Indo-American Conf. (KGSWC 2020)*, Mérida, Mexico, Nov. 2020, pp. 61–71.
[13] A. Mahindru and P. Singh, "Dynamic permissions based android malware detection using machine learning techniques," in *Proc. 10th Innovations Softw. Eng. Conf.*, Feb. 2017, pp. 202–210.
[14] S. Choi, S. Jang, Y. Kim, and J. Kim, "Malware detection using malware image and deep learning," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2017, pp. 1193–1195