INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 04, Issue 11, November 2024, pp : 2895-2903

www.ijprems.com
editor@ijprems.com

e-ISSN : 2583-1062
Impact Factor : 7.001

MALWARE DETECTION USING MACHINE LEARNING

Perla Vineetha[1]

[1]Institute/Organization: GMR Institute of technology

ABSTRACT

The shortcomings of traditional heuristic-based methods in malware detection are increasingly evident, as they are ineffective against rapidly evolving cyber threats. Heuristic approaches depend on static signatures, making it challenging to keep up with continuous changes and sophisticated evasion techniques employed by malware authors. To counter this, a behavior-based analysis strategy is employed, where malware is executed in a controlled sandbox environment to observe real-time actions, such as file modifications, system calls, and network activity. The collected behavioral data is transformed into sparse vector models, which are then classified using machine learning algorithms like Random Forest, Stochastic Gradient Descent, Extra Trees, and Gaussian Naive Bayes. These classifiers performed exceptionally well, achieving 100% accuracy, precision, and recall, demonstrating their robustness in malware detection. To further enhance the detection system, improvements such as data augmentation for greater diversity, deeper neural network architectures to capture complex patterns, regularization techniques to avoid overfitting, and hyperparameter tuning for optimal performance are proposed. Ensemble models are also suggested to boost accuracy and stability by combining the strengths of multiple classifiers. The effectiveness of these enhancements was validated using the CSIC 2010 HTTP dataset. This approach showcases the potential of combining behavior-based analysis with advanced machine learning, offering a powerful, adaptive, and automated solution for malware detection and positioning it as a formidable defense mechanism against sophisticated and evolving cybersecurity threats.

Keywords: Malware Detection, **CSIC 2010 HTTP Dataset,** Cyber Threats, Behavior-Based Analysis, Sandbox Environment, Neural Network Architectures.

1. INTRODUCTION

The rapid advancement of malware, including viruses, ransomware, and trojans, has become a major threat to both individuals and organizations. As these malicious programs grow more sophisticated, traditional cybersecurity measures are increasingly ineffective. Modern malware uses complex, evasive tactics to circumvent conventional defenses, creating an urgent need for robust and adaptive detection systems. Consequently, developing advanced malware detection frameworks has become a crucial research focus, emphasizing proactive solutions that can combat both known and emerging threats.

These advanced detection systems combine signature-based methods with anomaly detection powered by machine learning (ML) and artificial intelligence (AI). Signature-based approaches detect malware by identifying familiar patterns, but they fail to recognize new, unknown threats. To address this limitation, AI-driven anomaly detection models analyze extensive datasets to spot unusual and suspicious behaviors, thereby enhancing the ability to detect novel malware strains.

Several machine learning models are key to improving the accuracy of malware detection. Random Forest (RF) is a reliable ensemble model that constructs multiple decision trees, merging their predictions to increase accuracy and reduce overfitting, ensuring consistent performance even with complex datasets. Stochastic Gradient Descent (SGD) is an optimization technique that efficiently handles large-scale, high-dimensional data, making it suitable for extensive malware analysis. Extra Trees, a variation of decision trees, introduces randomness to improve generalization, enhancing the model’s adaptability to previously unseen threats. Gaussian Naive Bayes (Gaussian NB), while simple, is effective in scenarios where data features are normally distributed, providing fast and efficient classifications.

Achieving a balance between precision and recall is vital in deploying these models. Precision ensures the accurate identification of true threats while minimizing false positives, which is essential to prevent unnecessary alerts and maintain system efficiency. Recall focuses on maximizing threat detection to prevent malware from slipping through the cracks. Striking an optimal balance is crucial: too many false positives can overwhelm security teams and reduce efficiency, while false negatives can leave systems exposed to cyberattacks.

The research emphasizes the ever-evolving nature of cybersecurity, highlighting the necessity for continuous innovation to keep pace with evolving malware tactics. Integrating machine learning into cybersecurity frameworks can transform threat detection, and the study evaluates the effectiveness of current methods through case studies and experiments while exploring areas for future improvement.

Behavior-based analysis is a key focus of the study, which monitors the real-time actions and interactions of software to detect threats. Unlike traditional signature-based methods, behavior-based detection observes malware behavior to identify suspicious activity, offering a dynamic and adaptive approach to threat detection. When combined with machine learning, behavior-based analysis provides a comprehensive and real-time response to threats, addressing the limitations of static detection techniques.

In conclusion, merging behavior-based analysis with advanced machine learning models represents a significant advancement in malware detection. This combined strategy effectively tackles current cybersecurity challenges and is designed to adapt to future threats. The research emphasizes a multi-faceted approach that leverages the strengths of different classifiers, ensuring a resilient and adaptive defense. Continuous innovation is essential in this field, as the cyber threat landscape is constantly evolving. The study underscores the importance of machine learning and interdisciplinary collaboration to create secure digital environments. As malware and cyberattacks become more complex, proactive and comprehensive solutions are increasingly crucial. This research serves as a call for ongoing advancements to stay ahead in the battle against cyber threats.

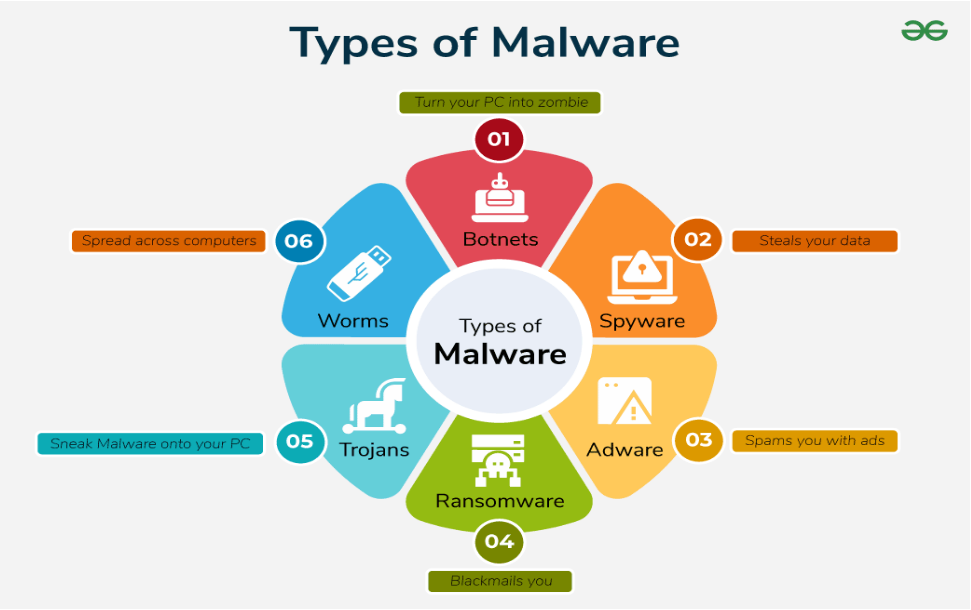


Fig: 1 Types of Malware

2. RELATED WORK

M.Shahpasand, L.Hamey and his team [1] proposed in 2019 methods to introduce vulnerabilities in the detection models of mobile malware against adversarial attacks. In this, they implement adversarial sample generation techniques for testing them by using samples against detection systems for revealing weaknesses and proposing strategies to enhance model robustness along with defense mechanisms. Zhao, J., Zhang and his team [2] has proposed in the year 2018, Implementation includes feature extraction of malware samples, and training models for detection, which presents better accuracy for detection and a certain robustness than static or dynamic features in their alone. Ahmad Mousa Altamimi, Maryam Al-Janabi [3] proposed, in 2020, doing a comparative analysis of the machine learning methods applied to classification and detection of malware. They implement and evaluate various algorithms over malware datasets on the basis of performance metrics; then they find which of these techniques is the most effective for accurate and efficient malware detection and classification. Bhatia, T., & Kaushal, R. [4] has proposed in the year 2017 explain malware detection in Android using dynamic analysis techniques. The implementation is on monitoring real time behavior of Android applications in a controlled environment that will eventually detect malicious activities. Their approach takes runtime features that has high detection accuracy since detection has based mainly upon patterns which cannot be unseen through just static analysis. Vatamanu, C., Cosovan, D and his team [5] has presented their work in the year 2015, which comprises works on signature-based, heuristic, and behavior-based malware detection techniques. In fact, their paper implements machine learning models such as decision trees and support vector machines by comparing their accuracy and effectiveness in malware detection with feature extraction and analysis. Chowdhury, M., Rahman, and his team [6] has introduced in the year 2017. Discusses the already prevailing machine learning techniques that are used for malware detection. Comparison of methods include decision trees, support vector machines, and neural networks. Authors have applied an enhanced model using support vector machine that analyses malware patterns and optimizes detection accuracy. Dhalaria, M., & Gandotra, E. [7] has proposed in the year 2020, the reviews on feature selection techniques and ensemble learning methods for Android malware detection, and it implements chi-square feature selection to reduce dimensionality, and it uses ensemble learning classifiers to improve the efficiency of detection performance with the higher accuracy as compared to other traditional methods. Gavriluţ, D., Cimpoeşu and his team [8] has proposed in the year 2009,This article discusses many approaches of machine learning to malware detection, including Bayesian networks and decision trees, deploys a machine learning-based system that extracts features from executable files, classifies benign versus malicious software using supervised classifiers, and hence leads to better detection rates. Gopaldinne, S. R., Kaur and his team [9]has proposed in the year 2021 suggested analysis of existing PDF malware classification techniques based on sign-ature-based and heuristic approaches. It gives a detailed overview of various classifiers that have a strong focus on machine learning-based approaches; furthermore, it presents an analysis of their performance for detecting malicious PDF files in consideration with feature extraction, and classification precision. Jamil, Q., & Shah [10] proposed in the year 2016 that compare various algorithms related to Naive Bayes, k-nearest neighbors and Support Vector Machines, to review Android malware detection techniques with the help of machine learning, and implemented a comparative analysis of these models by testing their performance and accuracy to detect Android malware through experimental results. Kuo, W. C., Liu [11] in 2019 has given a survey that analyzes hybrid malware detection methods by incorporating static and dynamic analysis techniques using machine learning and gives an implementation of such a detection model combining the techniques to extract features of Android applications and train classifiers for enhanced accuracy and robustness in identifying complex behavior of malware. Markel, Z., & Bilzor, M. [12] proposed in 2014 that built upon previous malware detection approaches based on machine learning by combining features from both static and dynamic analysis. The classifier was trained using a dataset of opcode sequences and API calls; the authors had optimized the selection of features to raise the detection rate compared with traditional signature-based methods. This easily helped in the identification of malware. Akhtar, M. S., & Feng, T.[13] suggested in 2023 that performance of various machine learning algorithms for malware detection can be compared and models like SVM, Random Forest, and Neural Networks. Techniques for feature extraction should be optimized as indicated in their work. So much improvement was overlaid by building this study over their evaluation. Optimized algorithms were implemented and tested for any improvements of precision in detection. Al Zaabi, A., & Mouheb, D. [14] recently proposed in 2020 that discussed an Android malware detection using static features such as permissions and API calls, using machine learning classifiers. This study is a follow-up of their research work with the addition of better feature engineering and testing other classifiers to enhance the robustness as well as efficiency of malware detection. P. P. P M and H. P. [15] proposed in the year 2022 that were created a PDF malware detection system using machine learning. Work improves feature selection methods and explores additional algorithms to increase detection capabilities.

COMPARISION TABLE

	Title	Year	Objective	Limitations	Advantages	gaps	Performance Metrics
1	Adversarial Attacks on Mobile Malware Detection Adversarial Attacks on Mobile Malware Detection	2018	crafting samples that evade detection while maintaining the functionality of the malware.	1.Model dependency 2.Evaluation constraints.	1.Novel Insights. 2.Practical Relevance.	1.Defense Mechanim	1.accuracy (99%)
2	Malware Detection Using Machine Learning Based on the Combination of Dynamic and Static Features.	2019	dynamic analysis and static analysis. This approach is designed to better handle the fast-changing nature of malware.	1.Data Imbalance 2. Resource Intensive	1.Scalabili 2.Combine Features.	1.Evasion Techniqe	1.precision 2.accuracy (99.6%)

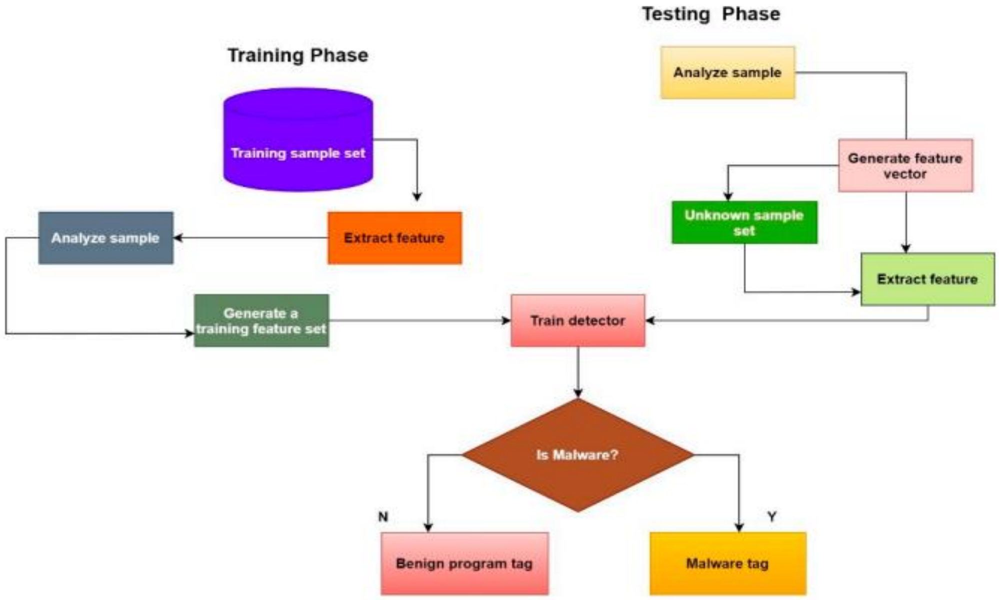
3	A Comparative Analysis of Machine Learning Techniques for Classification and Detection of Malware.	2020	Best feature extraction and classification methods that yield the highest accuracy in detecting malware.	1.Dataset Limitations 2.Scalability Issues	1.Practical Recommendation. 2.Identifyin-g Best Algorithms	1.Real-Time Detection	1.precison 2.false positive rates 3.accuracy (96%)
4	Malware Detection in Android based on Dynamic Analysis.	2017	conduct a comprehensive survey of machine learning techniques used in malware detection.	1.High Overhead. 2. Resource Intensive.	1.Behavior-Based Detection. 2.Comprehensive Feature Set.	1.Hybrid Analysis.	1.resource utilization 2.accuracy (96%)
5	A Comparative Study of Malware DetectionTechniqueUsing Machine Learning Methods.	2015	to evaluate the effectiveness of these techniques focusing on their ability to differentiate between benign and malicious software.	1. Dataset Constraints. 2. Resource and Time Requirement	1.Baseline for Future Research. 2.Contribut-ion to Best Practices.	1. Model Interpreta-bility. 2. Energy and Resource Efficiency	1.false negative rates. 2.accuracy (95%)
6	Protecting Data from Malware Threats using Machine Learning Technique.	2017	Protect data from malware threats by identifying and blocking malicious software before it can cause harm.	1. Data Dependency.2. Potential for Overfitting.	1. Adaptive Security. 2.Automate	1. Energy Efficiency2. Model Explainab-ility	1.precison 2.accuracy (95%)
7	Android Malware Detection using Chi-Square Feature Selection and Ensemble Learning Method.	2020	create a robust and efficient malware detection system to protect Android devices from malicious applications.	1.Overfitted2. Feature Selection Dependence.	1. Efficient Feature Selection. 2. Balanced Model Performace	1Adaptive Learning. 2. Model Interpreta-bility	1.recall 2.precison 3.accuracy (92%)
8	Malware Detection Using	2008	develop an automated,				1.accuracy

	Machine Learning.		adaptive system capable of accurately identifying and classifying malware, including previously unseen variants.	1. Data Quality and Availability.2. Static Analysis Constraints.	1.Automate and Efficiency. 2. Data-Driven Approach.	1. Model Explainab-ility. 2. Energy Efficiency	(99%)
9	Overview of PDF Malware Classifiers.	2021	develop an efficient, scalable, and automated system for detecting and classifying malicious PDF files.	1.Limited Coverage of Evasion Techniques. 2. Static Analysis Constraints.	1.Compreh-ensive Analysis. 2. Feature Extraction.	1.Evasion Resistant. 2Resource Optimization	1.accuracy (99.6%)
10	Analysis of machine learning solutions to detect malware in android.	2016	To evaluate different machine learning techniques for detecting Android malware, understand their strengths and weaknesses, and suggest ways to improve and adapt these methods for better malware detection.	Computation-al Resource Requirement	1Adaptable 2.Automate and Efficiency	1.Combin-ed the Static and Dynamic Analysis	1.precision 2.recall 3.features Utilization 4.accuracy (96%)
11	Study on android hybrid malware detection system based on ml.	2019	To evaluate and compare the performance of various machine learning models used in detecting Android malware, aiming to	1.Scalability Issues. 2. Data Dependency.	Feature Complete. Malware analysis	Explainab-ility	1.computati-onal time. 2.accuracy (90%).

			improve detection.				
12	Building a machine learning classifier for malware detection system.	2014	To explore the challenges and trade-offs involved in building and deploying machine learning models for malware detection.	1. Data Quality and Representativeness. 2.Overfitting Risk.	1. Feature-Rich Analysis 2.Automate Detection.	1.Hybrid Approach.2. Adaptive Learning	1.accuracy (85%-95%). 2.recall
13	Evaluation of machine learning algorithms for malware detection	2023	To analyze the impact of different feature extraction techniques on the performance of machine learning models.	1. Data Limitations 2. Model Interpretable	Comparative Analysis.	1. Hybrid Detection Technique	1.accuracy (85%-95%) 2.precison
14	Android malware detection using static features and machine learning.	2020	To develop and evaluate a machine learning-based approach for detecting Android malware using static analysis of application features.	Suscept to Evasion	1.Low Resource Usage. 2. Early Detection.	Handling Obfuscate Malware.	1.accuracy (85%-95%) 2.recall
15	PDF Malware detection based on machine learning .	2022	To design and implement a system for detecting malware in PDF files using machine learning techniques.	1.Limit Dynamic Analysis. 2.Data Dependency.	1.Improved Accuracy. 2.Scalabile	1.Real-Time Detection.2. Adaptive Learning Models.	1.accuracy (90%-95%) 2.recall 3.resource utilization

3. METHODOLOGY

Workflow Diagram



- The workflow emphasizes the two main phases: training and testing. In the training phase, the model learns from labeled data, and in the testing phase, it applies this knowledge to classify new, unlabeled data.
- Feature extraction is a critical step in both phases, as the quality and relevance of features greatly influence the model's accuracy.
- The decision-making process involves the trained detector making binary classifications: either the sample is malware or it is benign.
- The process is designed to automate the detection of malware using machine learning, reducing manual intervention and providing a scalable solution for identifying malicious software.

1. Training Phase:

- Training Sample Se**t**: The process begins with a dataset of samples, which consists of both benign and malicious files. This set is used for training the malware detection model.
- Analyze Sample: Each sample in the training dataset is analyzed to identify relevant features that can help differentiate between benign and malicious behavior.
- Extract Feature: Features are extracted from the analyzed samples, capturing characteristics such as code patterns, metadata, or specific behaviors relevant to malware identification.
- Generate a Training Feature Set: The extracted features are compiled into a structured dataset, known as the training feature set, which is used for model training.
- Train Detector: A machine learning algorithm is trained using the feature set. The algorithm learns to identify patterns and make predictions about whether a given sample is malware or not.

2. Testing Phase:

- Unknown Sample Set: New or unseen samples, which have not been used in the training phase, are introduced for testing the trained model.
- Analyze Sample: Similar to the training phase, each unknown sample is analyzed to prepare it for feature extraction.
- Extract Feature: Features are extracted from the unknown samples, creating a feature vector that serves as input to the trained detector.
- Generate Feature Vector: The extracted features are formatted into a feature vector, which the model can use to make a prediction.

3. Detection and Classification

- Train Detector: The trained malware detection model evaluates the feature vector and makes a decision.
- Decision Point ("Is Malware?"): The model determines whether the analyzed sample is malware or a benign program based on the learned patterns.
  - If **Yes (Y)**: The sample is tagged as malware.
  - If **No (N)**: The sample is tagged as a benign program.

Key steps that focus on detecting and analyzing malware using machine learning algorithms.

1.Dataset Collection and Preparation:

- The authors collected malware samples from VirusTotal and used a sandbox environment (Cuckoo Sandbox) to safely execute and record malware behavior.
- The dataset contained 373 samples (301 malware, 72 benign), and the behavior of these files was recorded in JSON reports.
- The features from these reports, such as system calls, network activity, and file modifications, were extracted and used as input for machine learning models.

2. Feature Extraction and Selection:

- From the collected data, features that are relevant for distinguishing malware from benign files were extracted and selected.
- By selecting the most important features, the authors reduced the data's complexity and improved the efficiency of the models.

3. Machine Learning Model Selection:

- The models tested in the study included Random Forest (RF), Stochastic Gradient Descent (SGD), Extra Trees, Gaussian Naive Bayes, k-Nearest Neighbors (KNN), Decision Tree (DT), and AdaBoost.
- These algorithms were chosen for their effectiveness in classification tasks and ability to handle large datasets with many features.

★ **Random Forest (RF):** It is a popular machine learning method used for classification and regression tasks. Random Forest is powerful because it combines the strengths of multiple trees to make reliable and accurate predictions.

★ **Stochastic Gradient Descent (SGD):** Stochastic Gradient Descent (SGD) is a way to train machine learning models by updating the model's parameters little by little. Instead of using the whole dataset at once, it makes updates using one data point at a time, which makes learning faster but noisier. This approach helps models learn efficiently, especially with large datasets.

★ **Extra Trees:** Extra Trees (Extremely Randomized Trees) is an ensemble machine learning technique that builds multiple decision trees by randomly selecting split points and features. It is similar to Random Forests but takes more randomness in both tree building and feature selection. Extra Trees is fast and often performs well on large datasets.

★ **Gaussian Naive Bayes:** Gaussian Naive Bayes is a classification algorithm that assumes features follow a normal (Gaussian) distribution. It calculates the probability of each class based on feature values and their distribution. It's simple, fast, and works well when the features are normally distributed.

★ **k-Nearest Neighbors (KNN):** K-Nearest Neighbors (KNN) is a classification algorithm that assigns a class to a data point based on the majority class of its nearest neighbors. It doesn't require training but calculates the distance between data points to classify them. KNN is simple and effective for small datasets but can be slow with large ones.

★ **Decision Tree (DT):** A Decision Tree (DT) is a model that makes decisions by splitting data into branches based on feature values. It uses a tree-like structure, where each node represents a feature, and each branch represents a decision rule. It's easy to understand, but can overfit if not carefully tuned.

★ **AdaBoost:** AdaBoost is an ensemble learning technique that combines multiple weak models (like decision trees) to create a strong model. It focuses on correcting the mistakes of previous models by giving more weight to misclassified data points. AdaBoost improves accuracy by iteratively adjusting weights and combining models.

4. Training and Testing the Models:

- The training process involved teaching the model to recognize patterns in the data that indicate malicious activity.
- After training, the models were tested on new, unseen data to assess their accuracy, precision, recall, and F1-score.
- The best-performing models, such as Random Forest and SGD, achieved perfect scores in all metrics.

5. Dynamic Malware Analysis:

- The study adopts dynamic analysis to detect malware, which means it observes the behavior of malware as it executes in a controlled environment (such as a sandbox).
- This method is preferred over static analysis because it is more difficult for malware to hide its behavior during execution.

6. Sandbox Environment (Cuckoo Sandbox):

- Malware samples were executed in a Cuckoo Sandbox, which is an isolated environment where malware behaviors are recorded without posing any risk to real systems.
- The sandbox captures malware behavior, such as system calls, network activity, and file modifications.

4. RESULTS AND DISCUSSION

Best approaches in malware detection are those that rely on machine learning techniques combining dynamic and static features with a high degree of accuracy greater than 90% [2][4][6][11]. Hybrid methods combining static code analysis with dynamic behavior monitoring enhance the detection rates through a combination of the best features offered by different techniques [10][14]. There are traditional techniques known as signature-based detection and pure static or dynamic analysis, forming the foundational capabilities, although these are not very effective against the constantly evolving threats [7][9][12]. Specifically, datasets like Malware Genome Project and Android-specific collections have to be used for training these models [5][13]. The prime objectives of these techniques are recognition and classification of malware, which prevent infections and thereby advance cybersecurity research [3][8][15].

5. CONCLUSION AND FUTURE DEVELOPMENT

The conclusion of the paper emphasizes the effectiveness of combining behavior-based analysis with machine learning for robust malware detection. Traditional heuristic methods have proven inadequate against rapidly evolving cyber threats, whereas the proposed approach, which uses sandbox execution and machine learning classifiers, demonstrated exceptional performance metrics, achieving 100% accuracy, precision, and recall. It highlights the adaptability of the behavior-based model and its significant potential in cybersecurity applications. To further enhance accuracy and system resilience, future advancements are recommended, including data augmentation to introduce variability and make the model more robust, as well as exploring deeper neural network architectures like CNNs and RNNs to capture complex malware behavior. Additionally, applying regularization techniques and leveraging hyperparameter optimization methods, such as Grid Search or Bayesian optimization, will help refine model performance. Ensemble learning strategies, such as stacking or boosting, are suggested to combine classifier strengths, while optimizing the system for real-time detection using online learning techniques is crucial. Expanding behavioral analysis to cover more sophisticated network patterns and integrating external threat intelligence feeds will ensure the system remains adaptive and formidable against highly evasive and evolving malware threats, thus positioning the research as a powerful and practical defense mechanism.

6. REFERENCES

[1] www.irjmets.com
[2] www.ijwer.com