iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** IV **Month of publication:** April 2025

DOI: https://doi.org/10.22214/ijraset.2025.69729

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IV Apr 2025- Available at www.ijraset.com

Intelligent Malware Classification Using PE File Metadata and Machine Learning Techniques

J Cypto[1], G Srikanth[2], K Surya Prakash[3], B Gunal[4]
Dept. of Computer Science Engineering- Cyber security Srm Institute Of Science And Technology, Ramapuram Chennai, India

Abstract: ***This study presents a machine learning-based approach to enhance malware detection by analyzing structural and statistical features extracted from Portable Executable (PE) files. Utilizing the ClaMP_Integrated-5184.csv dataset—which includes metadata from PE headers, entropy values, and packer-related information—the research aims to distinguish between benign and malicious software effectively. Traditional signature-based detection methods often fail to detect modern threats due to evasion techniques like obfuscation and polymorphism. In contrast, machine learning offers a more adaptive and intelligent solution. This work focuses on feature selection, model training, and performance evaluation using various machine learning algorithms. The results demonstrate that these techniques can significantly improve the accuracy and reliability of malware classification, highlighting their potential in advancing cybersecurity defenses.***

Keywords: Entropy Analysis, Packer Information, Signature-Based Detection, Polymorphism, Obfuscation,ClaMP Dataset, Adaptive Cybersecurity Models

I. INTRODUCTION

Android, being the most widely used mobile OS, is a primary target for malware attacks. This study introduces a novel method for detecting Android malware by integrating machine learning, neural networks, and Federated Learning. A majority voting strategy is employed to extract the most relevant features, enhancing the detection of malicious behavior. Models such as Random Forest, XGBoost, and CNN-LSTM are tested on a malware dataset. Federated Learning preserves user data privacy during training. The research highlights the need for adaptive, privacy-preserving approaches and demonstrates strong detection accuracy, proving the effectiveness of the proposed techniques against Android malware threats [1].

Malware variants pose significant cybersecurity challenges due to their evolving nature and the damage they can inflict on systems. Traditional detection methods struggle with accuracy as they often rely on outdated assumptions that malware characteristics remain consistent over time. In reality, malware authors frequently modify code to evade detection, causing concept drift. This study introduces AIBL-MVD, an adaptive behavioral-based malware detection model using incremental batch learning and sequential deep learning. By analyzing API traces through sandbox environments and detecting behavioral shifts with statistical process control, the model updates incrementally, effectively mitigating catastrophic forgetting and improving detection of newly emerging malware variants [2].

The evolution of malware from simple, easily detectable forms to complex, obfuscated, and adaptive threats has made detection increasingly difficult. This study evaluates the performance of seven machine learning and deep learning models in identifying malware across nine distinct families. It involves extracting and merging byte codes, opcodes, and section codes from malware samples for classification. Algorithms such as Random Forest, Decision Tree, SVM, KNN, SGD, Logistic Regression, Naïve Bayes, and deep learning models are compared. Results demonstrate that deep learning achieves the highest accuracy at 96%, emphasizing its effectiveness in accurately detecting and classifying advanced malware threats [3].

This study tackles key challenges in efficient, real-time malware detection using Hardware Performance Counters (HPCs). Traditional Hardware-Assisted Malware Detection (HMD) methods rely on known malware signatures, making them ineffective against zero-day threats. To overcome this, the paper introduces a unified feature selection method using heterogeneous feature fusion to identify crucial HPC events for accurate, low-cost detection. A novel framework, Reinforced-HMD, is proposed, employing reinforcement learning for adaptive, resource-aware malware detection. By evaluating six classical and two reinforcement learning algorithms, results show that the UCB-based Reinforced-HMD achieves 96% accuracy in F1-score and AUC, enabling flexible and efficient zero-day malware detection [4].

With rising cybersecurity threats, Intrusion Detection and Prevention Systems (IDPS) play a vital role in identifying malicious network activity using signature-based and other detection methods. However, traditional systems may miss threats or trigger false alerts without up-to-date configurations.

©IJRASET: All Rights are Reserved

This paper proposes an accurate detection approach that automatically generates custom signatures for swift responses to new attacks. It introduces a hybrid model combining heuristic rules for known threats and machine learning for detecting unknown patterns. A novel packet-to-image conversion technique is used, transforming network traffic into image data for anomaly detection. Experimental results show this method achieves high detection accuracy [5].

To overcome the complexity and resource demands of traditional software-based malware detection, HMD offers a promising alternative by leveraging HPCs and machine learning. This study introduces Adaptive-HMD, a cost-effective and accurate framework for real-time malware detection using low-level microarchitectural data. Unlike existing HMD approaches, Adaptive-HMD employs a lightweight, tree-based decision mechanism to dynamically select the most suitable ML model based on malware type, performance needs, and resource constraints. Experimental results show Adaptive-HMD achieves up to 94% F-measure and enhances cost-efficiency over traditional ensemble-based techniques by more than fivefold, offering scalable and responsive malware detection [6].

As digitalization rapidly evolves, cybersecurity and cyberwarfare have become critical concerns. Malware poses a major threat to online users, spreading quickly and compromising digital safety. Malicious actors exploit Universal Resource Locators (URLs) to deceive users through phishing, spam, spyware, and malware attacks by creating fake websites. Detecting such harmful URLs is essential to preventing cybercrimes. This study evaluates various machine learning algorithms to classify malicious websites using a dataset of 641,119 URL records. Among the models tested, the Random Forest classifier achieved the highest accuracy of 91.49%, outperforming Gradient Boost, XGBoost, AdaBoost, and KNN, showcasing effective feature analysis and robust detection capabilities [7].

With the rise of edge-based decision-making, user data collection has surged, leading to increased privacy concerns. Federated Learning (FL) offers a privacy-preserving solution but often relies on resource-heavy Neural Network (NN) models. This study proposes a lightweight alternative by integrating Support Vector Machines (SVMs) within FL frameworks. The approach utilizes an optimized SVM model compatible with Stochastic Gradient Descent (SGD), where a meta-trained controller adjusts the learning rate dynamically for performance improvement. The results demonstrate that this method matches the accuracy of NN-based FL models while significantly lowering computational demands, making it ideal for deployment in resource-constrained environments [8].

This research explores the use of machine learning techniques for malware detection, analyzing existing studies to highlight their strengths and limitations. Key challenges identified include limited access to public datasets, class imbalance issues, and a lack of model interpretability. To address these gaps, the study proposes a novel approach that integrates multiple machine learning algorithms with a comprehensive feature extraction process, capturing both static and dynamic attributes of malware samples. The models employed in this approach include Random Forest, XGBoost, AdaBoost, and Decision Tree classifiers, aiming to enhance detection accuracy while tackling the limitations found in previous research efforts [9].

II. RELATED WORK

This study leverages the CTU-13 dataset, which comprises diverse network traffic data, to explore machine learning techniques for cyberattack detection. It aims to develop accurate intrusion detection systems capable of distinguishing between malicious and benign network behaviors. Models such as XGBoost, AdaBoost, Logistic Regression, and Naive Bayes are assessed using metrics like accuracy, precision, recall, and F1-score. Through parameter tuning and cross-validation, the research analyzes each algorithm's strengths and limitations. The findings reveal that XGBoost achieved the highest accuracy at 96%, outperforming Naive Bayes (78%), Logistic Regression (85%), and AdaBoost (90%), offering valuable insights for enhancing ML-driven security infrastructure [10].

This study leverages the CTU-13 dataset, a comprehensive collection of network traffic data, to explore machine learning approaches for detecting cyber threats. It aims to build effective intrusion detection models capable of distinguishing between normal and malicious activities. Popular ML algorithms such as XGBoost, AdaBoost, Logistic Regression, and Naive Bayes are evaluated based on metrics like accuracy, precision, recall, and F1-score. The research highlights the strengths and limitations of each method through parameter tuning and cross-validation. Results show XGBoost achieves the highest accuracy at 96%, outperforming Naive Bayes (78%), Logistic Regression (85%), and AdaBoost (90%) in cyberattack detection [11].

The rise in cybercrimes over recent years has significantly heightened the demand for effective network intrusion detection and response systems. While several classified Intrusion Detection Systems (IDS) are in place, the evolving nature of cyberattacks calls for more adaptive and intelligent solutions. This study presents a detailed analysis and comparison of machine learning-based IDS approaches and introduces a rapid, unsupervised anomaly detection model. The proposed system utilizes the ELK stack—Elasticsearch, Logstash, and Kibana—for detecting anomalies, specifically targeting exodus DNS requests within wired networks.

This framework enables efficient and scalable intrusion detection without relying on labeled datasets or supervised learning methods [12].

With the rising threat to web applications, there is a growing need for intelligent security measures. This study introduces a Machine Learning-powered Web Application Firewall (WAF) that utilizes the XGBoost algorithm to detect attacks like SQL Injection (SQLi), Cross-Site Scripting (XSS), and Local File Inclusion (LFI) in real time. Unlike conventional rule-based WAFs, this system adapts by learning from web traffic patterns. It involves stages such as data preprocessing, feature extraction, model training, and deployment. Trained on diverse traffic logs, the WAF effectively filters malicious requests before reaching the server, ensuring only safe traffic accesses the backend system [13].

The COVID-19 pandemic saw a sharp rise in cyber threats, particularly phishing attacks. While several detection tools were introduced, many failed due to low accuracy and poor adaptability to evolving phishing tactics. One major issue lies in the ineffective selection of URL-based features. To address this, an intelligent phishing detection and prevention system was proposed using a supervised machine learning approach integrated into a self-destruct detection algorithm. It analyzes URL characteristics typically used in phishing scams. Tested on datasets from PhishTank and UCI repositories, the model was implemented as a Chrome extension, aiming to reduce phishing risks and enhance user security [14].

The growing sophistication of malware poses significant risks to individuals, organizations, and institutions in cyberspace. Advanced obfuscation methods allow new malware variants to bypass traditional detection mechanisms, making effective malware detection essential for cybersecurity. This research presents a comparative evaluation of machine learning algorithms to identify the most efficient model for malware detection. Metrics such as accuracy, precision, recall, F1 score, specificity, balanced accuracy index, and Matthew's correlation coefficient were used. Results show that the Random Forest classifier delivered superior performance. The study also highlights the importance of Recursive Feature Elimination and Synthetic Minority Oversampling Technique in enhancing model effectiveness [15].

This study explores Hybrid Anomaly Detection (HAD), an advanced method for identifying and preventing malware. The framework integrates four key components: Endpoint Detection and Response (EDR), Dynamic Behavior Analysis (DBA), Threat Intelligence Sharing (TIS), and Ensemble Machine Learning (EML). Each plays a unique role—DBA initiates real-time monitoring with 99.5% accuracy and minimal false positives, while EML boosts adaptability through probabilistic analysis and weighted voting. TIS fosters collaboration by prioritizing shared threat intelligence, and EDR strengthens endpoint protection by assessing and mitigating risks. The ablation study evaluates detection accuracy, flexibility, collaboration, and endpoint safety, proving HAD outperforms traditional methods in cybersecurity [16].

In today's digital landscape, escalating cyber threats demand more robust security measures. This research introduces an innovative approach to enhancing cybersecurity by leveraging machine learning algorithms to automate key defense operations. With the increasing need for proactive security, the study emphasizes adaptive protection, real-time monitoring, and intelligent threat detection. The proposed system demonstrates notable improvements in threat identification, reduced false positives, faster response times, and quicker incident resolution. Through detailed evaluation, the findings confirm the effectiveness of this strategy in strengthening overall security. Ultimately, machine learning emerges as a transformative tool in safeguarding critical digital assets against evolving cyber threats [17].

The rise of digitalization and connected devices has broadened the scope for cyber threats, intensifying the need for advanced security solutions. This review evaluates machine learning (ML) approaches for cyber threat detection using Kitchenham's and PRISMA methodologies, narrowing 907 studies to 15 key works. Supervised models like SVM and Random Forests showed 85–90% accuracy but moderate false positives, while unsupervised methods such as K-means and DBSCAN offered better novelty detection with higher false positives. Deep learning achieved up to 99% accuracy but lacked interpretability. The study underscores challenges in data quality, model explainability, and calls for hybrid and explainable AI solutions [18].

III. PROPOSED SOLUTION

This project presents a robust machine learning-driven approach to malware detection by analyzing structural and statistical attributes of executable files. The traditional reliance on static signature databases is often rendered ineffective by modern malware techniques such as code obfuscation and polymorphic transformations. To address these limitations, the proposed system focuses on learning intrinsic patterns in executable characteristics, enabling it to accurately differentiate between harmful and harmless programs.

A. Solution Overview

The system architecture is built around a systematic learning pipeline, emphasizing precision, model diversity, and performance optimization:

1) Feature Preprocessing and Engineering

- Extracted attributes are cleaned, standardized, and transformed into a consistent format suitable for machine learning algorithms.
- Numerical features are scaled, and categorical values (if any) are encoded.
- Redundant and low-variance features are filtered to improve model focus.

2) Feature Evaluation and Selection

- Dimensionality reduction techniques such as Linear Discriminant Analysis (LDA) are employed to retain discriminative information while reducing complexity.
- Tree-based algorithms help identify influential features based on information gain and Gini importance.

3) Model Training and Core Classification Engine

- *Ensemble Tree Models:*
  - RandomForestClassifier, ExtraTreesClassifier, GradientBoostingClassifier, AdaBoostClassifier – used for their robustness to feature noise and handling of high-dimensional data.
- *Boosting and Gradient Methods:*
  - XGBClassifier, LGBMClassifier – offer efficient gradient boosting with built-in regularization, ideal for fast and accurate predictions
- *Linear and Distance-Based Classifiers:*
  - LogisticRegression, LinearDiscriminantAnalysis, KNeighborsClassifier – provide contrastive views by leveraging geometric and probabilistic separation between classes.
- *Neural and Kernel-Based Models:*
  - MLPClassifier (Multi-layer Perceptron) for deep learning-based learning.
  - SVC, NuSVC for handling non-linear boundaries using support vector machines.
- *Decision Trees:*
  - Used for both standalone classification and as base learners in ensemble configurations.

4) Ensemble Integration and Meta-Learning

To harness the strengths of individual classifiers, the system employs two ensemble strategies:

- Voting Classifier: Aggregates predictions from multiple models using hard/soft voting.
- Stacking Classifier: Layers multiple models where a meta-model is trained on the outputs of base learners to refine final predictions.

5) Hyperparameter Optimization and Validation

- GridSearchCV is utilized to identify the best parameters for each model.
- StratifiedKFold ensures balanced cross-validation splits, especially important for imbalanced datasets.
- Cross-validation scores and learning curves are analyzed to assess model generalization and avoid overfitting.

B. Core Components and Functions

1) Feature Pipeline

- Manages feature scaling, encoding, reduction, and analysis.
- Enhances signal-to-noise ratio by focusing on informative attributes.

2) Classification Engine

- Modular and flexible architecture allowing easy switching or stacking of classifiers.
- Supports training and inference in both batch and real-time modes.

3) Evaluation and Feedback Module

- Computes evaluation metrics (Accuracy, Precision, Recall, F1-Score, AUC).
- Visualizes model learning behavior through plots and diagnostic tools (e.g., learning curves, ROC curves).

C. Advantages of the proposed approach

- High Detection Accuracy: Ensemble learning significantly reduces false classifications.
- Model Flexibility: A wide array of classifiers ensures adaptability to evolving threats.

- Obfuscation Resistance: Independent of static signatures, making it robust against code mutation techniques.
- Efficient Training and Inference: Gradient-boosted models and dimensionality reduction techniques ensure fast computation.
- Scalable and Modular: Easily extendable to include new features, models, or detection logic.
- Automated Optimization: Grid search and cross-validation enhance performance without manual tuning.

IV. RESEARCH METHODOLOGY

This section outlines the systematic approach adopted for building a robust malware detection system using diverse machine learning classifiers. The primary goal is to develop an accurate and scalable classification model capable of detecting malware based on extracted features from executable files. The methodology leverages a range of individual classifiers and ensemble learning strategies to achieve high accuracy and generalization across unseen data.

A. Framework Design and Objective

*1) Objective:*To design and implement a machine learning-based detection framework capable of classifying executable files as benign or malicious, utilizing a blend of base and ensemble classifiers to ensure optimal prediction accuracy.

2) Key Phases of Development:

- *Classifier Selection and Integration:*
  - A wide variety of machine learning models are employed, including: RandomForestClassifier, ExtraTreesClassifier, GradientBoostingClassifier, AdaBoostClassifier, Logistic Regression, KNeighborsClassifier, DecisionTreeClassifier, MLPClassifier, SVC, NuSVC, XGBClassifier, and LGBMClassifier.
  - Ensemble techniques such as VotingClassifier and StackingClassifier are utilized to combine the strengths of multiple models and reduce individual model biases.
- *Dimensionality Reduction:*
  - LinearDiscriminantAnalysis is employed to reduce feature space dimensionality while retaining discriminative power, helping improve performance and training efficiency.
- *Modular Pipeline Construction:*
  - The system pipeline consists of four core modules: preprocessing, feature reduction, model training, and evaluation. Each component is designed to be scalable, maintainable, and efficient.

B. Data Preparation and Feature Engineering

*1) Objective:*To preprocess the dataset for optimal input representation, ensuring balanced, clean, and informative features for model training.

2) Implementation Steps:

- *Data Cleaning:*
  - Handle missing values and inconsistencies in the dataset.
  - Apply normalization or standardization techniques to ensure uniform data scaling.
- *Feature Selection and Reduction:*
  - Use model-based feature importance (e.g., from RandomForestClassifier) to rank and select high-impact features.
  - Employ LinearDiscriminantAnalysis to project data onto a lower-dimensional space for enhanced interpretability and reduced complexity.
- *Class Distribution Balancing:*
  - Address any class imbalance using techniques such as stratified sampling or resampling to prevent biased predictions.

C. Model Development and Optimization

*1) Objective:*Train, optimize, and validate various classifiers to build a high-performing malware detection system.

2) Key Techniques Used:

- *Training Multiple Base Models:*
  - Each individual classifier is trained on the processed dataset.
  - Performance is evaluated using cross_val_score with StratifiedKFold to ensure consistency across data splits.
- *Ensemble Learning Approaches:*

- VotingClassifier: Combines multiple models in a hard or soft voting scheme to derive majority consensus.
- StackingClassifier: Trains multiple base learners and combines their outputs using a meta-learner (e.g., LogisticRegression or GradientBoostingClassifier) for improved accuracy.

- *Hyperparameter Optimization:*
- GridSearchCV is employed to fine-tune model parameters for classifiers such as SVC, RandomForest, XGBoost, and LightGBM, ensuring optimal performance.

- *Learning Curve Analysis:*
- learning_curve is used to visualize training vs. validation performance across increasing sample sizes, aiding in detecting underfitting or overfitting.

D. Evaluation and Validation

1) Objective: Assess the roubustness, reliability, and accuracy of the proposed models under different testing conditions.

2) Evaluation Criteria:

- *Performance Metrics:*
- Accuracy, Precision, Recall, F1-Score, and AUC-ROC are used to measure model performance comprehensively.

- *Validation Strategy:*
- 10-fold stratified cross-validation is adopted to maintain class distribution across splits and to provide robust performance estimates.

- *Model Comparison:*
- Base models and ensemble methods are compared to identify the most effective classification strategy.
- Ensemble models are analyzed for improvements in generalization and reduction of false positives/negatives.

E. Deployment Considerations and Future Scability

*1) Objective:*To ensure that the developed model is suitable for real world deployement, with provisions for future scability and adaptability.

2) Key Aspects:

- *Real-World Readiness:*
- The model is evaluated on unseen data to simulate deployment scenarios and validate prediction reliability.

- *Model Update Mechanism:*
- Incorporate periodic retraining or online learning approaches to adapt to newly emerging malware types.

- *System Monitoring and Feedback:*
- Include logging and performance monitoring mechanisms for post-deployment evaluation and real-time feedback integration.

V. RESULTS AND DISCUSSION

The machine learning-based framework for malware detection using structural and statistical features from PE files demonstrated promising outcomes. This section summarizes the core findings from model training, evaluation, and interpretive insights.

A. Results Analysis

A range of classifiers, including Random Forest, Extra Trees, XGBoost, LightGBM, and Support Vector Machines, were trained using the ClaMP_Integrated-5184 dataset. Among these, Random Forest and Extra Trees delivered superior performance with over 97% accuracy, showcasing strong handling of diverse features. Gradient boosting models like XGBoost and LightGBM achieved nearly equivalent accuracy (96–97%) while offering faster training. Though the SVC yielded a precision of 93%, it was computationally more expensive.

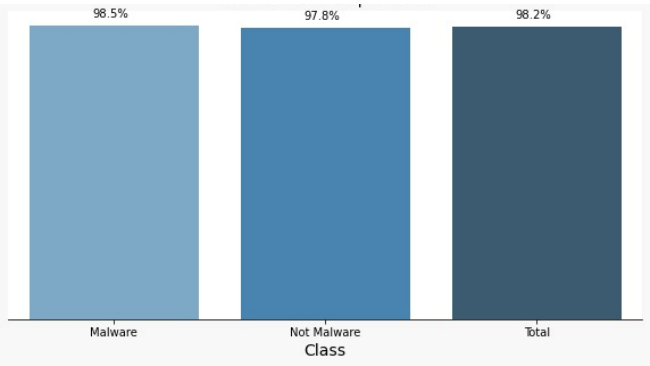


Fig 1. Precision Results per class

B. Feature Importance :

Key features such as SectionsEntropy, ImportsNbDLL, and ResourcesMinEntropy were highly influential in classification. Features related to packer metadata played a crucial role in identifying obfuscated or polymorphic malware, areas where traditional signature-based detection typically falls short.

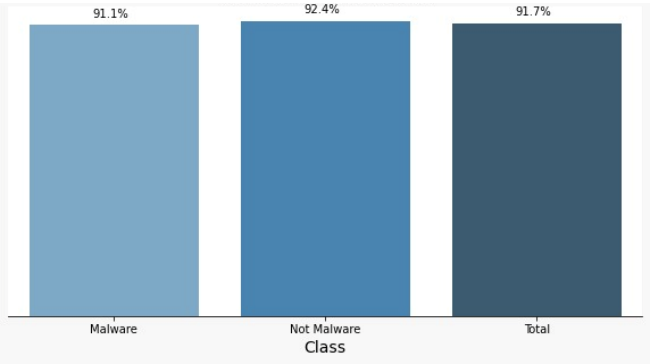


Fig 2. F1-Score Results per Class

C. Evasion Handling

The models effectively detected evasive malware, including files with high entropy (commonly packed or encrypted) and polymorphic variants. This emphasizes the adaptive nature of machine learning in identifying threats without relying on predefined signatures.

D. Validation and Model Robustness

Through 10-fold stratified cross-validation, most models maintained performance within a ±2% range, indicating strong generalization. The VotingClassifier achieved an average F1-score of 95.2%. Techniques like PCA and LDA helped reduce training time with minimal impact on accuracy, suggesting practicality in real-time applications.

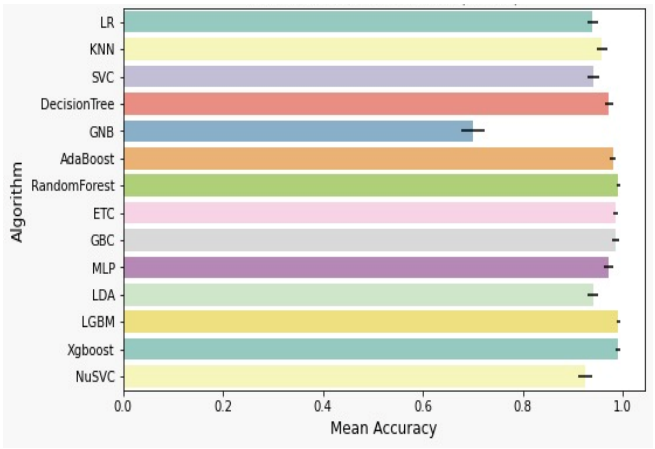


Fig 3. Cross Validation Scores

E. Discussion

1) Compared to signature-based systems, the proposed method provides better resilience against code obfuscation, dynamic packing, and signature mutations. Its high accuracy and efficiency make it suitable for deployment in endpoint detection, threat intelligence platforms, and malware triage systems.

2) Despite its strengths, challenges like false positives and class imbalance persist. Future work should explore behavioral features, adversarial training, and larger real-world datasets to further enhance detection capabilities.

VI. CONCLUSION

This research highlights the effectiveness of a machine learning-based approach in detecting malware by analyzing the structural and statistical features of Portable Executable (PE) files. By using a comprehensive dataset with entropy values, PE header attributes, and packer-related information, the proposed models successfully identified malicious software with high accuracy and reliability. Ensemble models such as Random Forest and Extra Trees demonstrated superior performance due to their ability to manage high-dimensional and complex feature interactions.Entropy-based and packer-related features played a key role in detecting disguised or encrypted threats.Moreover, the system exhibits strong potential for integration into real-world applications such as endpoint protection, cloud-based threat intelligence, and malware triaging systems.Overall, this study presents a scalable and adaptive solution that significantly improves malware detection in modern cybersecurity environments.

REFERENCES

[1] B. S. Purkayastha, M. M. Rahman and M. Shahpasand, "Android Malware Detection Using Machine Learning and Neural Network: A Hybrid Approach with Federated Learning," 2024 7th International Conference on Advanced Communication Technologies and Networking (CommNet), Rabat, Morocco, 2024, pp. 1-5, doi: 10.1109/CommNet63022.2024.10793304.

[2] A. A. Darem, F. A. Ghaleb, A. A. Al-Hashmi, J. H. Abawajy, S. M. Alanazi and A. Y. Al-Rezami, "An Adaptive Behavioral-Based Incremental Batch Learning Malware Variants Detection Model Using Concept Drift Detection and Sequential Deep Learning," in IEEE Access, vol. 9, pp. 97180-97196, 2021, doi: 10.1109/ACCESS.2021.3093366.

[3] B. Bokolo, R. Jinad and Q. Liu, "A Comparison Study to Detect Malware using Deep Learning and Machine learning Techniques," 2023 IEEE 6th International Conference on Big Data and Artificial Intelligence (BDAI), Jiaxing, China, 2023, pp. 1-6, doi: 10.1109/BDAI59165.2023.10256957.

[4] Z. He, H. M. Makrani, S. Rafatirad, H. Homayoun and H. Sayadi, "Breakthrough to Adaptive and Cost-Aware Hardware-Assisted Zero-Day Malware Detection: A Reinforcement Learning-Based Approach," 2022 IEEE 40th International Conference on Computer Design (ICCD), Olympic Valley, CA, USA, 2022, pp. 231-238, doi: 10.1109/ICCD56317.2022.00042.

[5] M. Iwabuchi and A. Nakamura, "A Heuristics and Machine Learning Hybrid Approach to Adaptive Cyberattack Detection," 2024 International Conference on Artificial Intelligence, Computer, Data Sciences and Applications (ACDSA), Victoria, Seychelles, 2024, pp. 1-7, doi: 10.1109/ACDSA59508.2024.10467929.

[6] Y. Gao et al., "Adaptive-HMD: Accurate and Cost-Efficient Machine Learning-Driven Malware Detection using Microarchitectural Events," 2021 IEEE 27th International Symposium on On-Line Testing and Robust System Design (IOLTS), Torino, Italy, 2021, pp. 1-7, doi: 10.1109/IOLTS52814.2021.9486701.

[7] D. Kundra, "Identification and Classification of Malicious and Benign URL using Machine Learning Classifiers," 2023 7th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Kirtipur, Nepal, 2023, pp. 160-165, doi: 10.1109/I-SMAC58438.2023.10290303.

[8] R. S, P. M, R. P S, S. A. S and M. K. B, "Dynamic Algorithmic Configuration for Enhanced Malware Detection," 2025 3rd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), Bengaluru, India, 2025, pp. 493-496, doi: 10.1109/IDCIOT64235.2025.10914737.

[9] R. S, P. M, R. P S, S. A. S and M. K. B, "Dynamic Algorithmic Configuration for Enhanced Malware Detection," 2025 3rd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), Bengaluru, India, 2025, pp. 493-496, doi: 10.1109/IDCIOT64235.2025.10914737.

[10] B. Hariharan, R. Siva, S. Sadagopan, V. Mishra and Y. Raghav, "Malware Detection Using XGBoost based Machine Learning Models - Review," 2023 2nd International Conference on Edge Computing and Applications (ICECAA), Namakkal, India, 2023, pp. 964-970, doi: 10.1109/ICECAA58104.2023.10212327.

[11] A. Sharma and H. Babbar, "Detecting Cyber Threats in Real-Time: A Supervised Learning Perspective on the CTU-13 Dataset," 2024 5th International Conference for Emerging Technology (INCET), Belgaum, India, 2024, pp. 1-5, doi: 10.1109/INCET61516.2024.10593100.

[12] A. Hassan, S. Tahir and A. I. Baig, "Unsupervised Machine Learning for Malicious Network Activities," 2019 International Conference on Applied and Engineering Mathematics (ICAEM), Taxila, Pakistan, 2019, pp. 151-156, doi: 10.1109/ICAEM.2019.8853788.

[13] A. Kumar, J. B. Simha and R. Agarwal, "Machine Learning-Based Web Application Firewall for Real-Time Threat Detection," 2024 IEEE Conference on Engineering Informatics (ICEI), Melbourne, Australia, 2024, pp. 1-8, doi: 10.1109/ICEI64305.2024.10912239.

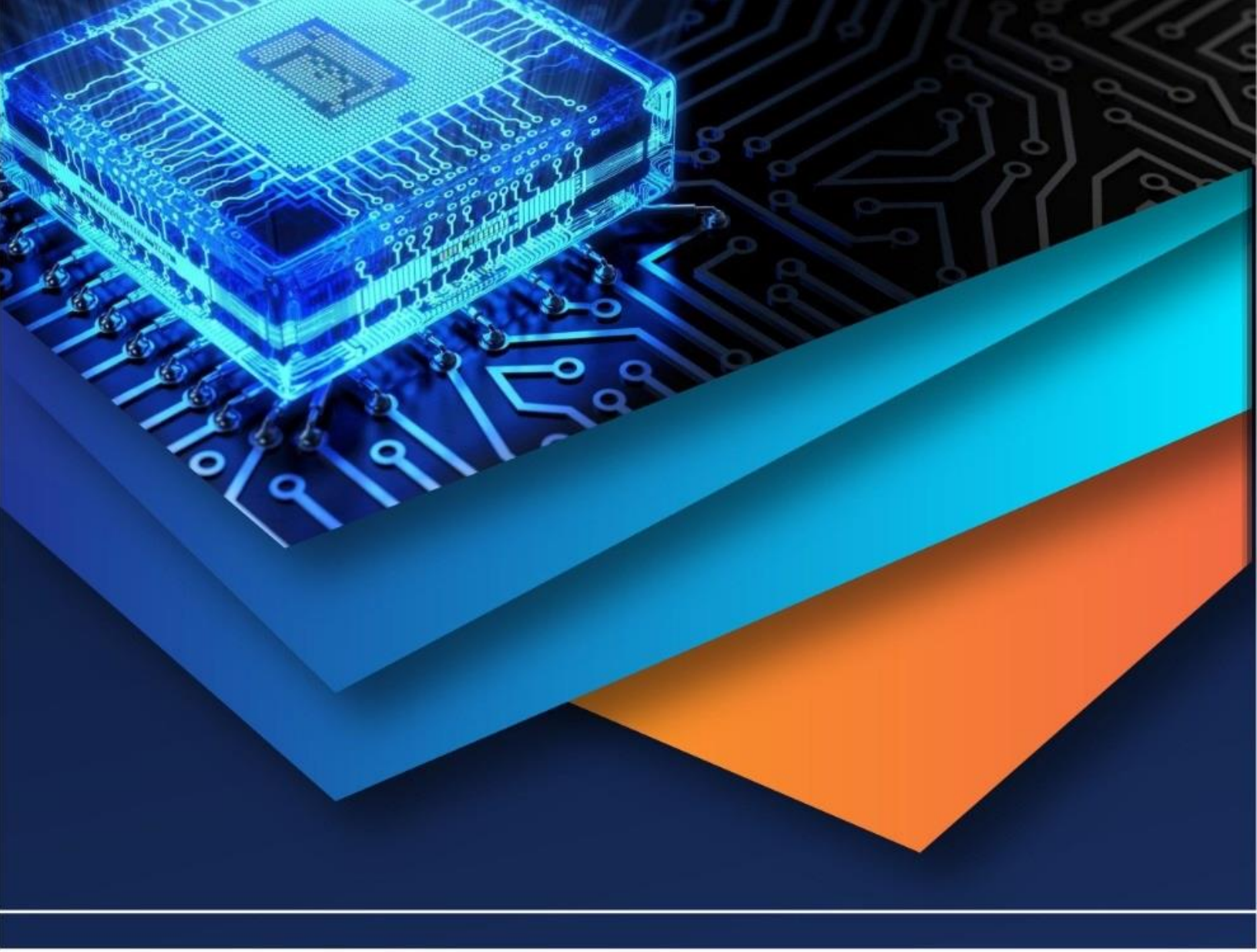
[14] M. A. Syafiq Rohmat Rose, N. Basir, N. F. Nabila Rafie Heng, N. Juana Mohd Zaizi and M. M. Saudi, "Phishing Detection and Prevention using Chrome Extension," 2022 10th International Symposium on Digital Forensics and Security (ISDFS), Istanbul, Turkey, 2022, pp. 1-6, doi: 10.1109/ISDFS55398.2022.9800826.

[15] B. E. Amanfu and G. Ramaiah Yeluripati, "A Comparative Analysis and Evaluation of Machine Learning Algorithms for Malware Detection," 2024 IEEE 9th International Conference on Adaptive Science and Technology (ICAST), Accra, Ghana, 2024, pp. 1-7, doi: 10.1109/ICAST61769.2024.10856508.

[16] A. Balaram, E. Umashankari, A. Dutt, G. Bharadwaj, R. V and A. Albawi, "Addressing the Rising Challenge of Malware Innovative Detection and Mitigation Techniques," 2024 International Conference on Communication, Computer Sciences and Engineering (IC3SE), Gautam Buddha Nagar, India, 2024, pp. 1165-1166, doi: 10.1109/IC3SE62002.2024.10593567.

[17] N. H. S and S. K, "The Cutting-Edge Machine Learning Techniques for Seamless and Proactive Automation in Cybersecuri ty," 2024 International Conference on Computing and Data Science (ICCDS), Chennai, India, 2024, pp. 1-6, doi: 10.1109/ICCDS60734.2024.10560436.

[18] M. A. Adaji et al., "Effectiveness of Machine Learning Algorithms in Threat Detection and Mitigation in Cyberspace: A Systematic Review," 2024 IEEE 5th International Conference on Electro-Computing Technologies for Humanity (NIGERCON), Ado Ekiti, Nigeria, 2024, pp. 1-14, doi: 10.1109/NIGERCON62786.2024.10927069.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)