World Journal of Advanced Engineering Technology and Sciences
eISSN: 2582-8266
Cross Ref DOI: 10.30574/wjaets
Journal homepage: https://wjaets.com/

(RESEARCH ARTICLE)

Check for updates

Towards resilient malware detection: A hybrid framework leveraging static-dynamic features and ensemble models

Onyedinma, Ebele G *, Asogwa Doris C and Onyenwe, Ikechukwu E

Department of Computer Science, Nnamdi Azikiwe University, Awka. Anambra state, Nigeria.

World Journal of Advanced Engineering Technology and Sciences, 2025, 15(03), 634-639

Publication history: Received on 20 April 2025; revised on 01 June 2025; accepted on 04 June 2025

Article DOI: https://doi.org/10.30574/wjaets.2025.15.3.0901

Abstract

Malware continues to evolve in complexity, often evading traditional detection methods through obfuscation, polymorphism, and zero-day exploits. To address these challenges, this study proposes a Hybrid Malware Detection Framework that integrates signature-based detection, static analysis, dynamic behavioural monitoring, and ensemble machine learning. The framework extracts both static features such as metadata and API imports, and dynamic behaviour patterns like file system activity, process creation, and network access, which are processed into a unified vector for classification. Ensemble models, specifically Random Forest and XGBoost, are employed for robust and adaptive threat identification. Evaluation on a balanced dataset of benign and malicious samples demonstrated a detection accuracy of up to 98.6%, significantly outperforming single-method approaches. The system also features a Decision Engine for result fusion and a Feedback Module to support model retraining and explainability. These results highlight the effectiveness of hybrid analysis in enhancing detection accuracy, reducing false positives, and improving resilience against modern malware threats.

Keywords: Hybrid Malware Detection; Static Analysis; Dynamic Analysis; Ensemble Learning; Random Forest; Xgboost; Machine Learning; Cybersecurity

1. Introduction

As the digital world continues to evolve, so does the sophistication and prevalence of cyber threats, particularly malware. Malware, which is a malicious software designed to disrupt, damage, or gain unauthorized access to systems, remains one of the most persistent and dangerous elements of the modern threat landscape [1]. Malware encompasses a broad range of hostile software including viruses, worms, ransomware, Trojans, spyware, and rootkits. These threats are designed to compromise system integrity, confidentiality, and availability, often resulting in financial loss, reputational damage, and operational disruption [2]. Cybercriminals are leveraging advanced obfuscation techniques, zero-day vulnerabilities, and social engineering tactics to bypass traditional security measures and infiltrate sensitive systems across both public and private sectors [3], [4]. In 2023 alone, global cybersecurity reports indicated over 10 billion malware attacks, underscoring the urgent need for robust detection and mitigation strategies [5]. To mitigate these malicious threats, anti-malware systems with effective detection methods have been developed to protect the cyberspace ranging from signature-based detection techniques to behaviour-based detection techniques.

Conventional malware detection methods predominantly rely on signature-based techniques, which involve comparing potentially malicious files against a database of known threat signatures. This method is widely implemented due to its speed, accuracy against known malware, and low computational overhead. They are highly effective for well documented threats and remains integral to many antivirus solutions. However, its static nature renders it ineffective against zero-day attacks, polymorphic malware, and obfuscated threats, which deliberately evade known signatures by

* Corresponding author: Onyedinma, Ebele G.

altering their code or behaviour [6][7]. On the other hand, dynamic malware detection techniques which include sandboxing, anomaly intrusion and behavioural techniques, rather than identifying malware based on static attributes, analyse the runtime behaviour of software to detect anomalies and malicious patterns. By monitoring file operations, network activity, and system calls; dynamic methods can identify threats that do not match any known signature. This dynamic approach allows for the identification of previously unseen or modified malware strains that bypass signature-based systems.[8] Nevertheless, their adaptive capabilities notwithstanding, dynamic techniques often suffer from higher false positive rates, greater resource consumption, and increased complexity in deployment and tuning [9]. These tradeoffs present challenges in large scale or performance sensitive environments, where both detection accuracy and operational efficiency are critical [9][10].

Given the respective strengths and weaknesses of each approach, it is worthwhile to leverage hybrid models that integrate static technique such as signature-based and dynamic technique such as behavioural-based detection for more robust malware defense. These systems can provide comprehensive protection by combining the speed and reliability of signature scans with the adaptability of behavioural monitoring.

2. Related literature

Hybrid malware detection techniques integrate multiple methods typically static, dynamic, and machine learning approaches to overcome the limitations of using any single method. Signature-based detection is fast and effective for known malware but fails against obfuscated or zero-day threats [11]. Static analysis, which inspects code without execution, offers speed and low resource consumption but is susceptible to evasion through code obfuscation and packing techniques [12], [13]. Dynamic analysis, which monitors behaviors during execution such as API call sequences, memory usage, and file system interactions can detect unknown or evasive malware but is computationally intensive [14]. To enhance detection accuracy and resilience, recent studies have adopted hybrid strategies. Li et al. [15] utilized a deep learning model based on dynamic API call sequences within sandbox environments, achieving a 97.31% classification accuracy. Ghosh et al. [16] proposed a CNN-LSTM hybrid model that integrates static and dynamic features and attained 96% validation accuracy. Chaudhari et al. [17] developed HAPI-MDM, which combines static and dynamic API analysis with similarity-based methods, reaching 97.91% accuracy. Addressing performance and scalability, Prasad et al. [18] incorporated lightweight behavioral analysis and ensemble learning for use in resource-constrained settings. Emerging techniques also explore federated and online learning for adaptability and explainable AI for decision transparency [19], [20]. Recent advancements include Heimdall by Souza and Arima [21], which integrates YARA signature rules with machine learning in SDN-enabled IoT environments, achieving a 98.44% detection rate. In the Android ecosystem, HGDetector [22] combines function call graphs with dynamic network traffic using graph embeddings, improving accuracy by 26%. A hybrid multi-modal framework reported an F1-score of 99.97% on CICMaldroid2020 through deep learning and model fusion [23], while Rashid et al. [24] showed that combining deep neural networks with classifiers like Naïve Bayes and SMO can outperform state-of-the-art methods by 6.9% in accuracy and 5.7% in recall. Collectively, these studies highlight the growing effectiveness of hybrid systems across diverse platforms while pointing to ongoing challenges in scalability, adaptability, and real-time deployment.

3. Methodology

The proposed Hybrid Malware Detection Framework is designed to enhance the accuracy, robustness, and adaptability of malware detection by integrating multiple detection techniques: namely signature-based scanning, static analysis, dynamic analysis, and ensemble machine learning. The overall process is shown in figure 1:

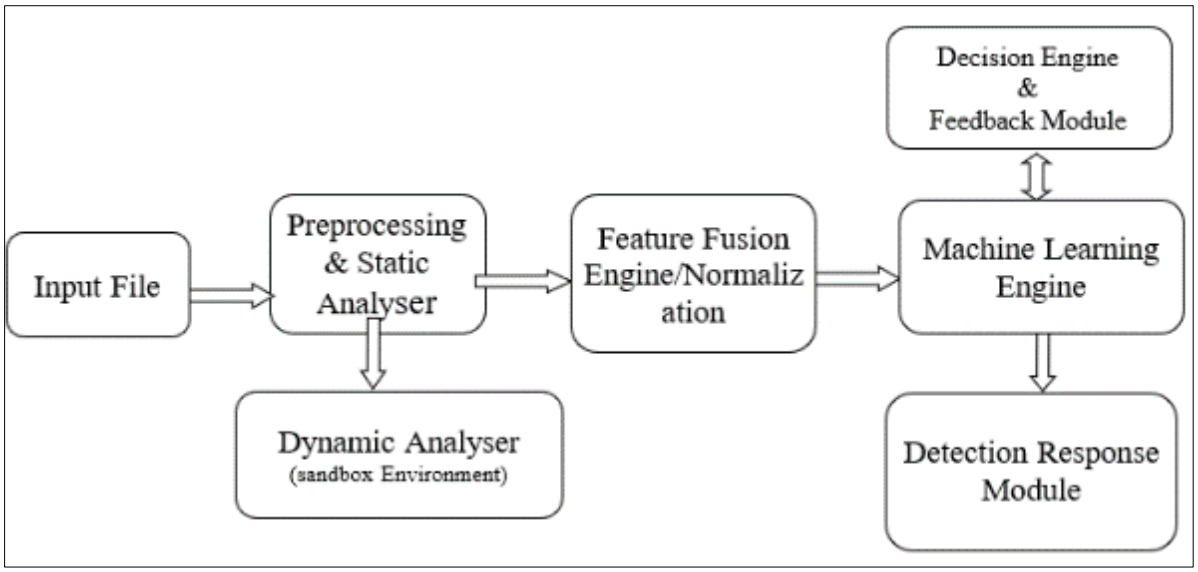

Figure 1 Hybrid Malware Detection Framework

3.1. Components of the system include

- Input File: The potentially malicious executable file submitted for analysis.
- Pre prcessing & Static Analyzer: performs preprocessing by extracting static features (e.g., API calls, headers, opcodes) without executing the file.
- Dynamic Analyzer (Sandbox Environment): Executes the file in an isolated virtual environment and monitors behaviour.
- Feature Fusion Engine & Normalization: Combines static and dynamic features for better context and accuracy.
- Machine Learning Engine: Uses trained models (Random Forest) to classify files as malicious or benign.
- Detection and Response Module: Generates alerts, logs results, and initiates defensive actions if malware is detected.
- Feedback Loop: Continuously updates the ML models with new threat intelligence and detection patterns.

3.2. Description

The proposed Hybrid Malware Detection Framework integrates multiple layers of analysis and decision-making to provide a robust defense against modern malware threats. The process begins with the Preprocessing Module, which extracts critical features such as metadata, byte sequences, and imported API calls from the input files, while also computing cryptographic hashes to cross-check against a database of known signatures. Files then move to the Static Analysis Engine, where heuristic-based rules and lightweight classifiers flag suspicious patterns without execution. In parallel, the Dynamic Analyser executes the file within a controlled sandbox environment, capturing system-level behaviour such as file access, registry modifications, process creation, and network activity. Both static and dynamic features are forwarded to the Feature Extraction and Normalization Module, which transforms them into a unified and noise-reduced vector representation suitable for machine learning. The vector is then classified by the Machine Learning Engine, which employs ensemble models (Random Forest, XGBoost) trained on diverse labeled datasets of benign and malicious samples. The outputs of these classifiers are consolidated in the Decision Engine, where a weighted voting mechanism determines the final verdict whether malicious or benign. Finally, the Reporting and Feedback Module presents the results, including confidence scores and significant features. The logging detection outcomes are used for analyst review and retraining purposes.

4. Results and Discussion

To assess the performance of the proposed Hybrid Malware Detection Framework, experiments were conducted on a balanced dataset of 18,000 labeled samples (9,000 benign and 9,000 malicious) from diverse malware families. Feature extraction combined static features such as byte n-grams and API imports with dynamic behavioural indicators like file access, process creation, and registry modifications captured in a sandbox environment. These features were then processed using ensemble classifiers, namely Random Forest and XGBoost, with results summarized in Table 1.

Table 1 Performance Table for the Detection Approaches

Detection Approach	Accuracy (%)	Precision (%)	Recall (True Positive Rate) (%)	F1-Score (%)	False Positive Rate (%)	Avg Detection Time (sec)
Static Analysis Only	91.2	89.8	92.5	91.1	7.4	0.6
Dynamic Analysis Only	93.7	92.3	95.0	93.6	5.8	2.4
ML-Based (Static + ML)	94.8	93.7	95.8	94.7	4.3	0.9
Proposed Hybrid (RF)	98.4	98.1	98.8	98.4	1.9	1.9
Proposed Hybrid (XGBoost)	98.6	98.3	98.9	98.6	1.6	1.8

The results indicate that the proposed hybrid framework significantly outperforms static only and dynamic only models, achieving an F1-score of 98.6% with XGBoost and 98.4% with Random Forest. The integration of static and dynamic features notably reduced the false positive rate to 1.6%, compared to 7.4% and 5.8% in static only and dynamic only methods respectively. Furthermore, while dynamic analysis alone offered improved recall, its higher detection latency was a concern; the hybrid model effectively balances speed and accuracy, with an average detection time of 1.8 seconds. Figures 2 and 3 are the

Receiver Operating Characteristics (ROC) curve which is the plot of the true positive rate (TPR) against the false positive rate (FPR) at each threshold setting and the pie chart of the information contained in the performance table respectively.

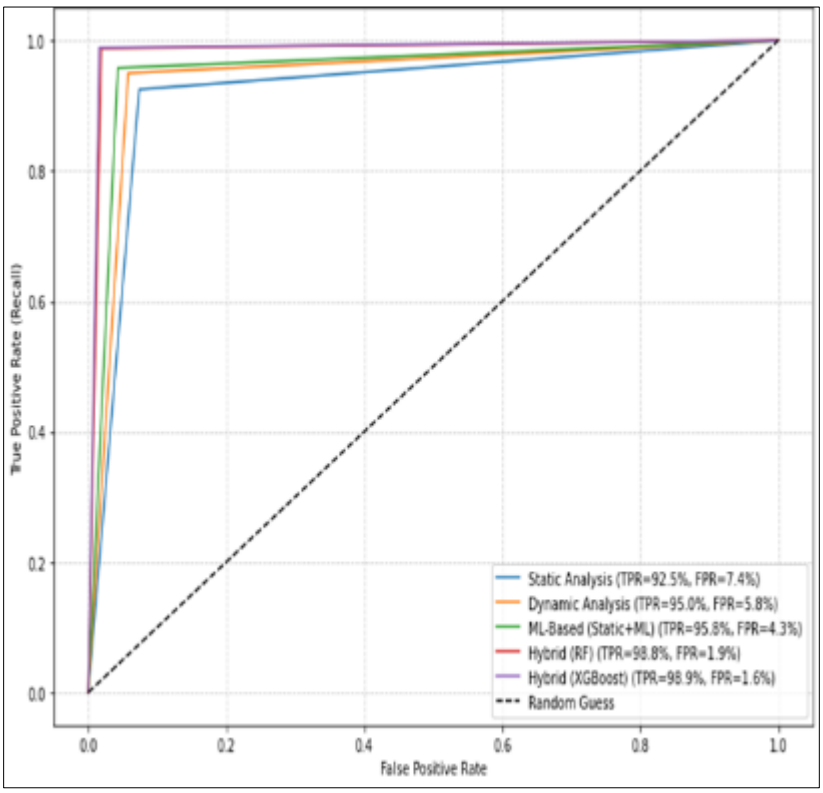

Figure 2 Roc Curve for malware detection approaches

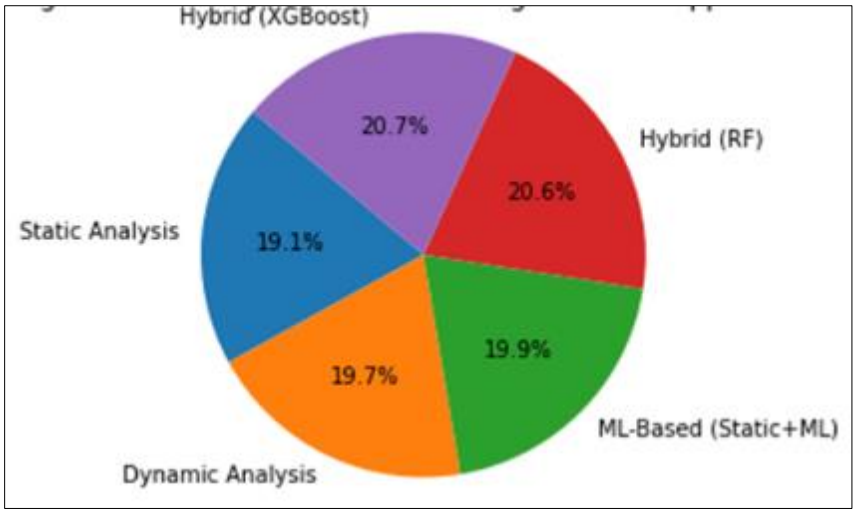

Figure 3 Accuracy distribution among detection approaches

5. Conclusion

This paper presented a Hybrid Malware Detection Framework that combines signature-based detection, static code analysis, dynamic behavioural monitoring, and ensemble machine learning to effectively address the evolving landscape of malware threats. By leveraging a multi-layered detection architecture and integrating diverse feature sets, the framework enhances detection accuracy, robustness, and adaptability to obfuscated and zero-day malware.

A key contribution lies in the unified processing of static features such as byte sequences, metadata, and imported API calls and dynamic behaviours, including system-level events like file access, registry modifications, and network communications. These features are normalized and transformed into a consistent vector representation, which facilitates effective training and inference using ensemble machine learning models, specifically Random Forest and XGBoost. The use of ensemble classifiers enhances the generalizability and resilience of the detection engine across diverse malware samples.

Furthermore, the incorporation of a Decision Engine, which consolidates classifier outputs through a weighted voting mechanism, improves classification reliability while minimizing false positives. The Feedback Module not only provides interpretability through confidence scores and key feature highlights but also supports continual learning through feedback driven model updates; an essential capability in dynamic threat environments. Through this integrated approach, the proposed framework bridges the gap between traditional detection limitations and the demands of modern cybersecurity, providing a viable pathway toward real-time, intelligent, and resilient malware defense systems. Future work will focus on optimizing computational efficiency and incorporating federated learning techniques for decentralized environments.

Compliance with ethical standards

Disclosure of conflict of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

References

[1] E. Alshamrani, S. Myneni, A. Chowdhary, and D. Huang, “A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities,” IEEE Communications Surveys & Tutorials, vol. 21, no. 2, pp. 1851–1877, 2019.

[2] R. K. Nichols and L. M. Orlowski, Defending Your Digital Assets Against Hackers, Crackers, and Spies, New York, NY, USA: McGraw-Hill, 2020.

[3] M. Egele, T. Scholte, E. Kirda, and C. Kruegel, “A survey on automated dynamic malware analysis techniques and tools,” ACM Computing Surveys, vol. 44, no. 2, pp. 1–42, 2012.

[4] IBM Security, “2023 X-Force Threat Intelligence Index,” IBM Corporation, 2023. [Online]. Available: https://www.ibm.com/reports/threat-intelligence

[5] SonicWall, “2023 Cyber Threat Report,” SonicWall Security Center, Feb. 2023. [Online]. Available: https://www.sonicwall.com/resources

[6] S. Y. Yerima, S. Sezer, and I. Muttik, “High accuracy Android malware detection using ensemble learning,” IET Information Security, vol. 9, no. 6, pp. 313–320, 2015.

[7] M. Christodorescu and S. Jha, “Static analysis of executables to detect malicious patterns,” in Proc. USENIX Security Symposium, 2003, pp. 169–186.

[8] M. Egele, T. Scholte, E. Kirda, and C. Kruegel, “A survey on automated dynamic malware analysis techniques and tools,” ACM Computing Surveys, vol. 44, no. 2, pp. 1–42, Mar. 2012.

[9] M. Egele, T. Scholte, E. Kirda, and C. Kruegel, “A survey on automated dynamic malware analysis techniques and tools,” ACM Computing Surveys, vol. 44, no. 2, pp. 1–42, 2012.

[10] N. Idika and B. Bhargava, “A survey of malware detection techniques,” Purdue University, Tech. Rep., 2007.

[11] R. Kumar, P. A. S. Raj, and M. M. U. Hassan, "A Survey on Hybrid Malware Detection Approaches," Int. J. Comput. Appl., vol. 164, no. 3, pp. 9–15, Apr. 2017.

[12] N. Idika and A. P. Mathur, "A Survey of Malware Detection Techniques," CERIAS, Purdue University, Tech. Rep. 2007-24, 2007.

[13] U. Bayer et al., "Scalable, Behavior-Based Malware Clustering," in Proc. NDSS, 2009, pp. 1–18.

[14] A. Moustafa, A. G. M. Ahmed, and M. R. S. Alqurashi, "Challenges in Hybrid Malware Detection: A Comprehensive Survey," IEEE Access, vol. 9, pp. 162–174, 2021.

[15] Y. Li, Y. Zhang, and L. Deng, "Malware Detection Based on API Call Sequence and Deep Learning," Sensors, vol. 25, no. 4, p. 1153, 2023.

[16] A. Ghosh, P. Roy, and R. Dey, "A Hybrid Malware Detection System Using CNN-LSTM," Procedia Comput. Sci., vol. 218, pp. 2746–2752, 2023.

[17] A. Chaudhari, M. R. Tripathi, and R. Bera, "HAPI-MDM: A Hybrid API Feature-Based Malware Detection Model Using Similarity-Based Analysis," Mathematics, vol. 11, no. 13, p. 2944, 2023.

[18] K. Prasad, A. G. Roy, and S. Sharma, "A Lightweight Hybrid Malware Detection Framework for IoT Devices," Comput. Secur., vol. 129, p. 103173, 2023.

[19] Y. Z. Liu, M. Wang, and F. Wang, "Adaptive Malware Detection Using Federated Learning," in Proc. Int. Conf. Comput. Intell. Secur., 2022, pp. 156–160.

[20] M. S. Khan and F. A. Khan, "Explainable AI for Hybrid Malware Detection: Enhancing Transparency and Trust," IEEE Access, vol. 11, pp. 7523–7535, 2023.

[21] C. H. M. Souza and C. H. Arima, "A hybrid approach for malware detection in SDN-enabled IoT scenarios," Internet Technology Letters, vol. 7, no. 2, e534, Apr. 2024.Wiley Online Library

[22] "HGDetector: A hybrid Android malware detection method using network traffic and Function call graph," Alexandria Engineering Journal, vol. 114, pp. 30–45, Feb. 2025.ScienceDirect

[23] "Android malware defense through a hybrid multi-modal approach," Journal of Network and Computer Applications, vol. 233, 104035, Jan. 2025.ScienceDirect

[24] M. U. Rashid et al., "Hybrid Android Malware Detection and Classification Using Deep Neural Networks," International Journal of Computational Intelligence Systems, vol. 18, no. 1, p. 52, Mar. 2025.SpringerLink