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 1, October 2025
Impact Factor: 7.67

AI Threat Detection Malware Analysis

Siddhant Mundre[1], Prof. D. G. Ingale[2], Prof. A. P. Jadhao[3], Prof. S. V. Raut[4], Prof. R. N. Solanke[5]
Student, CSE, Dr. Rajendra Gode Institute of Technology and Research, Amravati, India[1]
Guide, CSE, Dr. Rajendra Gode Institute of Technology and Research, Amravati, India[2]
Mentor, CSE, Dr. Rajendra Gode Institute of Technology and Research, Amravati, India[3,4,5]

Abstract: *This paper examines the application of artificial intelligence (AI) and machine learning (ML) techniques for threat detection and malicious software (malware) analysis. As cyber threats escalate in volume and sophistication, conventional signature-driven defences struggle against polymorphic and zero-day attacks. AI-powered methods — spanning static, dynamic and hybrid analysis — bring adaptability, pattern recognition, and automation to cybersecurity operations. The manuscript surveys contemporary literature, evaluates prevailing approaches, identifies limitations such as adversarial evasion and dataset bias, and proposes a hybrid framework combining static feature extraction, behavioural dynamic analysis, and an adversarially-hardened ensemble of deep learning and interpretable models. Empirical guidance for dataset curation, evaluation metrics, and deployment considerations is offered. The paper concludes with prospective directions including threat-intelligence integration, federated learning for privacy-preserving detection, and model explainability to enhance forensic utility. This research aims to furnish practitioners and researchers with a consolidated yet practical reference for advancing AI-driven malware defences*

Keywords: Artificial Intelligence; Machine Learning; Malware Analysis; Threat Detection; Deep Learning; Static Analysis; Dynamic Analysis; Adversarial Robustness; Explainable AI; Intrusion Detection

I. INTRODUCTION

The expanding attack surface of modern computing environments — including cloud infrastructure, mobile devices, Internet-of-Things (IoT) endpoints and supply-chain components — has accelerated the need for intelligent defense mechanisms. Traditional anti-malware systems rely heavily on signature matching and rule-based heuristics, which are fast but brittle against unseen or obfuscated threats. Machine learning and deep learning methods introduce the ability to generalize from data and detect previously unseen attacks by recognizing anomalous patterns, instruction sequences, system-call behaviors, or network traffic signatures.

This paper focuses on the synthesis of AI techniques employed in threat detection and malware analysis, emphasizing how static, dynamic, and hybrid pipelines are constructed and evaluated. We synthesize findings from recent survey articles, empirical evaluations and industry reports to highlight strengths and failure modes of existing methods. In doing so, we aim to present a reproducible methodological pathway for building robust, production-capable AI defenses that can be integrated into Security Operations Centers (SOC) and endpoint detection and response (EDR) platforms.

Motivation: Attackers increasingly use automated tools, code polymorphism, and AI-assisted evasion to craft malware that can bypass static scanners. Defensive AI must therefore be resilient, interpretable, and adaptive. This motivates hybrid approaches that combine feature-level explainability with the expressive capability of deep models and the operational safety of ensemble decision-making.

II. LITERATURE SURVEY

Research into ML-driven malware detection spans decades but has intensified with the advent of deep learning and increased availability of diverse datasets. Surveys in recent years classify approaches into static (binary and source-code feature extraction), dynamic (execution-trace and behavior monitoring), and hybrid categories that combine both sources.

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 1, October 2025
Impact Factor: 7.67

Static analysis uses features such as opcode sequences, byte-level n-grams, import/export tables, header metadata and control-flow graphs. Early ML efforts applied decision trees, support vector machines (SVMs) and Random Forests to engineered features. Deep learning approaches later transformed raw binaries into images, used NLP-inspired embeddings of opcode sequences, or modeled call-graphs with graph neural networks.

Dynamic analysis inspects runtime behavior: system calls, network connections, file and registry access patterns, and memory snapshots. Sequence models (LSTM/GRU), attention-based transformers, and trace-embedding networks are used to classify or cluster behaviors. Dynamic traces are resilient to superficial obfuscation but require controlled execution environments (sandboxes) and can be resource-intensive.

Hybrid techniques merge static and dynamic representations, often improving detection rates and reducing false positives. More recent literature emphasizes adversarial robustness, explainability (XAI), and the importance of standardized benchmarks. Industry reports also underscore that automated tooling adoption is high among organizations, but adversaries are increasingly capable of evading defenses, necessitating continuous model retraining and threat-intelligence integration. Notable recent surveys and systematic reviews document these trajectories and call for reproducible datasets and interpretable models. (See referenced surveys in References.)

III. EXISTING WORK

Classical machine learning models (SVM, Random Forests, Naive Bayes) remain competitive on curated datasets and often deliver efficient inference for constrained devices. Deep convolutional networks and recurrent models have been applied to both raw byte streams and behavioural traces, achieving higher detection rates at the cost of complexity and compute.

Graph-based approaches model program structure (call-graphs, data-flow graphs) using Graph Neural Networks (GNNs), providing robustness to certain code transformations. Transformer-based architectures, adapted from NLP, have been used to embed opcode sequences and API call logs, benefiting from pretraining on large corpora of benign and malicious binaries.

Adversarial machine learning is an active subfield: attackers craft perturbations and obfuscations that cause misclassification, and defenders respond with adversarial training, input sanitization, and detection-of-adversarial-example subsystems. Explainable AI techniques (SHAP, LIME, attention visualization) have been applied to provide human-readable rationales for alerts, which is crucial for SOC analyst triage.

Industry tools increasingly combine signature-based rules with ML scoring to reduce alert volume. Endpoint Detection and Response (EDR) and Network Traffic Analysis (NTA) solutions feed telemetry into centralized ML models for cross-host correlation and prioritization. Despite maturation, open challenges remain: dataset bias, reproducible benchmarking, runtime efficiency, and the arms race with adversarial attackers.

IV. METHODOLOGY

This section outlines a practical pipeline for an AI-driven malware detection and analysis system, designed to be robust, interpretable, and deployable.

1. Data Collection and Labeling:

- Obtain diverse datasets: benign software, commodity malware families, polymorphic samples, and modern ransomware. Sources should include public repositories (e.g., VirusShare, VirusTotal where permitted), vendor telemetry, and synthetic samples.
- Labeling must be multilabel-aware: family attribution, behavior tags (ransomware, info-stealer), and confidence metadata from multiple AV engines.
- Address class imbalance using stratified sampling, weighted loss functions, or synthetic minority oversampling.

2. Feature Engineering:

- Static features: header metadata, import/export tables, n-gram sketches of bytes, opcode embeddings, and control-flow graph fingerprints.

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 1, October 2025
Impact Factor: 7.67

- Dynamic features: system-call sequences, network fingerprints, file-system operations, registry modifications, and memory entropy changes.
- Cross-modal features: time series of resource usage, correlated with network indicators and parent-process lineage.

3. Model Design:

- Ensemble architecture combining:
  a) Lightweight static classifier (Random Forest) for quick triage.
  b) Deep sequence/transformer model for behavioral traces.
  c) GNN for structural code representations.
  d) Meta-classifier that fuses outputs with confidence calibration (e.g., Platt scaling or isotonic regression).
- Incorporate adversarial training by injecting obfuscation transformations and gradient-based adversarial examples during training.

4. Explainability & Triage:

- Use SHAP value approximations and attention maps to highlight contributing features.
- Generate succinct investigation notes for SOC analysts indicating root-cause hypotheses and recommended remediation.

5. Evaluation:

- Metrics: precision, recall, F1-score, ROC-AUC, PR-AUC, time-to-detect, and false positive rate (FPR).
- Use temporal validation splits and cross-environment evaluation to measure generalization.
- Benchmark against signature-only baselines and ablate ensemble components to quantify contribution.

V. PROPOSED WORK

We propose a hybrid, adversarially-hardened ensemble framework named HADES (Hybrid-Adversarial Detection Ensemble System), which integrates the following:

Architecture:

- Static Triage Module: a fast Random Forest classifier using header and n-gram features to provide immediate risk scoring.
- Behavioural Deep Module: a transformer-based model trained on sandbox traces and system-call sequences, pre-trained with contrastive objectives to improve representation quality.
- Structural Graph Module: a GNN that models function-call graphs and inter-procedural relationships, capturing program semantics resilient to linear code obfuscations.
- Fusion & Adversarial Hardening: a meta-learner that uses calibrated probabilities and uncertainty estimates (e.g., Monte Carlo dropout) and is trained with adversarial augmentations.

Key Innovations:

-Contrastive pretraining on unlabeled telemetry to produce robust embeddings that improve few-shot detection of novel families.
- Federated updates: allow organizations to contribute model updates without sharing raw telemetry, preserving privacy while improving cross-organization detection capability.
- Explainable alarms: automated generation of analyst-facing rationales and a ranked list of indicators (IOCs) extracted from both static and dynamic analyses.

Dataset & Experimental Plan:

- Curate a mixed dataset of ~200k samples combining benign executables, known malware families and synthetic polymorphic variants.
- Evaluate HADES against baselines (signature engine, single-model ML, and existing hybrid models) using cross-validation, temporal holdouts, and adversarial attack simulations.

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 1, October 2025
Impact Factor: 7.67

- Measure not only detection metrics but operational cost: compute, memory, and average time-to-triage for SOC workflows.

Security Considerations:

Implement continuous monitoring for model drift and automated retraining triggers. Harden the training pipeline against poisoning by monitoring data provenance, using robust aggregation, and differential privacy techniques for shared updates.

VI. CONCLUSION

Artificial intelligence offers transformative capabilities for threat detection and malware analysis, improving detection of novel and evasive threats when combined with careful engineering and operational practices. Hybrid architectures that fuse static, behavioral and structural representations achieve stronger generalization and can reduce false positives when coupled with explainability modules for analyst triage.

Nevertheless, challenges persist: adversarial evasion, dataset biases, compute constraints at the edge, and the need for standardized, reproducible benchmarks. The proposed HADES framework demonstrates a pragmatic path forward by unifying multiple representation modalities, pretraining strategies, adversarial hardening, and privacy-preserving collaboration.

Future work should prioritize federated learning experimentation, standardized evaluation suites, and integration of threat-intelligence feeds to enable predictive, rather than purely reactive, defenses. Collaborative information sharing, combined with rigorous privacy safeguards, will be critical to scale AI-driven security across diverse organizations.

REFERENCES

[1]. Bensaoud, "A Survey of Malware Detection Using Deep Learning," arXiv:2407.19153, 2024.
[2]. S. Salem et al., "Advancing cybersecurity: a comprehensive review of AI-driven methods," Journal of Big Data, 2024.
[3]. Gopinath, "A comprehensive survey on deep learning based malware detection," 2023.
[4]. SANS Institute, "2024 Detection and Response Survey," 2024.
[5]. WindowsCentral reporting on AI-powered malware evasion (Black Hat disclosures), 2025.
[6]. Ferdous J., "A Survey of ML Techniques for Multi-Platform Malware," MDPI, 2025.