International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 3, May-Jun-2026, ISSN (Online): 2395-566X

Ai-Based Cybersecurity Using Machine Learning For Real-Time Threat Detection: A Review Of Techniques, Datasets And Deployment Architectures

Chakave Sakshi Revannath[1]; Prof. M. A. Auti[2]; Prof. M. C. Kolse[3]; Prof. S. K. Said[4]
Department Of Artificial Intelligence And Data Science Jaihind College Of Engineering Pune, Maharashtra, India

Abstract—The number and sophistication of cyberattacks, including malware, phishing, ransomware and denial-of-service campaigns, continue to grow, and signature-based security tools struggle to recognize new or modified threats. Machine Learning (ML) offers a complementary approach that learns the characteristics of normal and malicious behaviour from data and flags suspicious activity in real time. This paper reviews ML-based threat detection with emphasis on network intrusion detection. It traces the evolution from the classical intrusion-detection model to rule-based systems, and examines supervised classifiers such as Decision Trees, Random Forests and Support Vector Machines, unsupervised anomaly detectors and deep learning models. Public benchmark datasets, feature engineering from network flows, evaluation metrics and the practical difficulties of class imbalance, concept drift and false alarms are discussed. A layered architecture for a real-time detection system is proposed, comprising traffic capture, feature extraction, a hybrid detection engine, alert management and response integration. A research gap analysis, objectives and a comparative analysis of algorithms are presented. The review concludes that ensemble ML models combined with explainable alerts and continuous retraining can significantly improve detection accuracy and reduce false alarms in operational security environments.

Index Terms—Cybersecurity, intrusion detection, machine learning, random forest, support vector machine, decision tree, anomaly detection, real-time threat detection

I. INTRODUCTION

Organizations depend on networks for every business process, and attackers exploit this dependence through malware, phishing, ransomware, brute-force attempts and distributed denial-of-service attacks. Traditional defences such as firewalls and signature-based intrusion detection systems (IDS) are effective against known threats but are blind to novel attacks for which no signature exists.

Denning's intrusion-detection model [1] proposed detecting abnormal use of a system by comparing observed behaviour against statistical profiles, an idea that underlies modern ML-based detection. Machine learning can generalize from examples, adapting to variations of known attacks and highlighting unusual behaviour that may indicate zero-day threats.

This paper reviews ML approaches to real-time threat detection, focusing on network traffic analysis with Decision Tree, Random Forest and Support Vector Machine classifiers. It proposes a deployment architecture, identifies research gaps and discusses challenges in moving from laboratory accuracy to operational value.

II. RELATED WORK

Buczak and Guven [2] surveyed data-mining and ML methods for intrusion detection and found that tree ensembles and SVMs were among the most accurate, while emphasizing that results depend heavily on the dataset used. Tavallaee *et al.* [3] analysed the widely used KDD Cup 99 dataset, revealed a large number of redundant records that inflate accuracy, and released the cleaner NSL-KDD subset.

Because older datasets do not reflect modern traffic, Moustafa and Slay [4] created UNSW-NB15 with contemporary attack families, and Sharafaldin *et al.* [5] produced CICIDS2017 with realistic background traffic and labelled flows for attacks such as brute force, botnets, web attacks and DDoS. These datasets are now standard benchmarks.

Sommer and Paxson [6] cautioned that anomaly detection in operational networks is harder than other ML applications because of the high cost of errors, the diversity of benign traffic and the difficulty of obtaining labelled data; they recommended narrowly scoped detectors and careful evaluation. In response, Mirsky *et al.* [7] proposed Kitsune,

© 2026 IJSRET

an online ensemble of autoencoders that learns normal traffic without labels and runs on modest hardware in real time.

Deep learning has also been explored. Shone *et al.* [8] combined stacked non-symmetric autoencoders with a Random Forest, Yin *et al.* [9] used recurrent networks for multi-class detection, and Vinayakumar *et al.* [10] compared deep neural networks with classical classifiers across several datasets and proposed a scalable hybrid framework. Liu and Lang [11] reviewed these trends and noted that explainability and real-world validation remain weak points. Table I summarizes the surveyed works.

TABLE I Summary of Literature Survey

Ref.	Method	Key Findings	Limitations
[1]	Statistical intrusion-detection model	Foundation of anomaly-based detection	Profiles hard to maintain at scale
[12]	Snort signature-based IDS	Lightweight, widely deployed rule engine	Cannot detect unknown attacks
[3]	Analysis of KDD Cup 99; NSL-KDD	Removed redundant records inflating accuracy	Traffic still outdated
[2]	Survey of ML / data mining for IDS	Compared methods; trees and SVM strong	Results dataset-dependent
[6]	Critical analysis of ML for IDS	Explained why anomaly detection is hard in practice	No new detector proposed
[5]	CICIDS2017 dataset	Realistic labelled flows for modern attacks	Some labelling errors reported
[7]	Kitsune online autoencoder ensemble	Unsupervised, real-time on modest hardware	Sensitive to training-period contamination
[8]	Non-symmetric autoencoder + Random Forest	Improved accuracy with reduced training time	Evaluated on older datasets
[10]	Deep neural networks vs. classical ML	Scalable hybrid IDS framework	Limited explainability

III. PRELIMINARIES

1. Types Of Intrusion Detection

Signature-based detection matches traffic against known attack patterns, as in the open-source Snort system [12]; it has few false positives but cannot detect new attacks. Anomaly-based detection models normal behaviour and flags deviations; it can detect novel attacks but may raise more false alarms. Hybrid systems combine both. Fig. 1 classifies ML techniques used for detection.

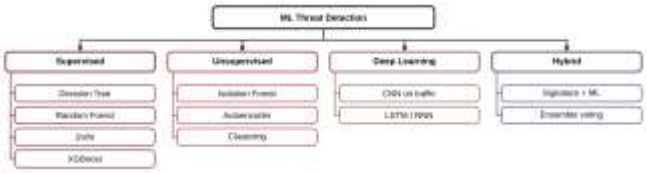


Fig. 1. Classification of ML techniques for threat detection.

2. Algorithms

A Decision Tree recursively splits data on the feature that best separates classes, producing human-readable rules. A Random Forest [13] trains many trees on random subsets of data and features and combines their votes, which improves accuracy and reduces over-fitting. A Support Vector Machine [14] finds the hyperplane that separates classes with the largest margin and can model non-linear boundaries through kernels. Gradient boosting such as XGBoost [15] builds trees sequentially to correct earlier errors.

IV. PROBLEM STATEMENT AND OBJECTIVES

1. Identified Research Gaps

The following gaps are identified:

- Many reported accuracies come from outdated or redundant datasets and do not transfer to live networks.
- Severe class imbalance causes models to miss rare but dangerous attacks.
- Models degrade as traffic patterns change (concept drift), yet retraining strategies are rarely evaluated.
- Alerts are often unexplained, making triage slow for security analysts.
- Integration of ML detectors with response workflows and threat frameworks such as MITRE ATT&CK [16] is limited.

2. Objectives Of The Review

The objectives of this review are:

- To study signature, anomaly and hybrid approaches to threat detection.

© 2026 IJSRET

- To compare Decision Tree, Random Forest, SVM and deep learning models on accuracy, speed and interpretability.
- To review benchmark datasets and evaluation metrics for intrusion detection.
- To propose a real-time, layered detection architecture with explainable alerts.
- To identify challenges such as false alarms and drift, and suggest mitigation strategies.

3. Proposed Framework

The proposed architecture in Fig. 2 processes traffic as a stream and produces prioritized, explained alerts.

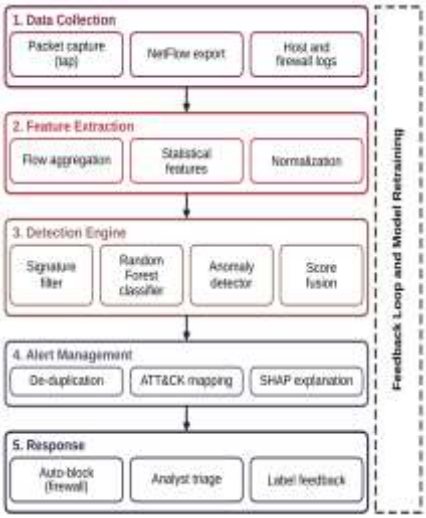


Fig. 2. Proposed real-time ML-based threat detection architecture.

4. Data Collection Layer

Packets are captured from network taps or switch mirror ports, and flow records are exported from routers. Host logs and firewall events are collected in parallel.

B. Feature Extraction Layer

Packets are aggregated into bidirectional flows, and statistical features such as duration, packet counts, byte counts, inter-arrival times, flag counts and protocol type are computed. Features are normalized using parameters learned during training.

5. Detection Engine

A signature module first removes known threats. Remaining flows are scored by a Random Forest classifier for known attack categories and by an unsupervised anomaly detector for unusual behaviour. Scores are combined into a final threat score.

6. Alert Management Layer

Alerts are de-duplicated, enriched with context, mapped to MITRE ATT&CK techniques and accompanied by SHAP-based [17] explanations that show which features drove the decision.

7. Response Layer

High-confidence alerts can trigger automated actions such as blocking an IP address at the firewall, while lower-confidence alerts are queued for analyst review. Analyst decisions are fed back as labels for retraining.

V. WORKING METHODOLOGY

The detection workflow appears in Fig. 3. Model performance is assessed using precision, recall and F1-score rather than accuracy alone, because benign traffic dominates:

$$Precision = TP / (TP + FP), \quad Recall = TP / (TP + FN) \quad (1)$$

$$F1 = 2 \cdot Precision \cdot Recall / (Precision + Recall) \quad (2)$$

The false-positive rate FP/(FP+TN) is monitored closely, since even a small rate produces many false alarms at high traffic volumes. Class imbalance is handled with class weighting or synthetic oversampling during training.

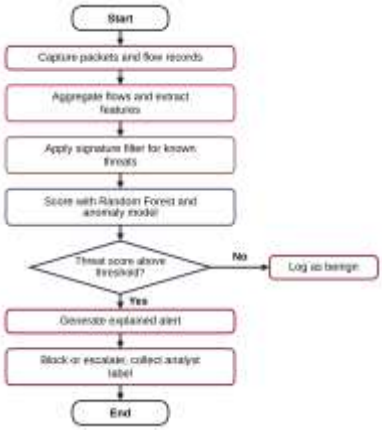


Fig. 3. Real-time threat detection workflow.

Detection latency is measured from the end of a flow to alert generation; tree-based models typically score a flow in microseconds, making them suitable for real-time use.

VI. DISCUSSION

1. Comparison Of Approaches

Table II compares the algorithms commonly used for threat detection. Random Forest offers the best balance of accuracy, speed and robustness; Decision Trees are the most interpretable; SVMs perform well on smaller datasets but scale poorly; and deep models can learn complex patterns but require more data and computation.

© 2026 IJSRET

TABLE II Comparison of Detection Algorithms

Algorithm	Accuracy	Speed	Interpretability
Decision Tree	Medium–High	Very fast	High
Random Forest	High	Fast	Medium
SVM	High (small data)	Slow on large data	Low
XGBoost	High	Fast	Medium
Deep learning	High	Moderate (GPU)	Low
Isolation Forest	Medium (unknowns)	Fast	Medium

2. Open Challenges And Future Work

Table III summarizes key challenges. Future research directions include federated learning across organizations to share threat knowledge without sharing raw traffic, detection of attacks within encrypted traffic using metadata, adversarially robust models that resist evasion, and large language models that help analysts summarize and investigate alerts. Alignment with the NIST Cybersecurity Framework 2.0 [18] can help organizations integrate ML detection into broader risk management.

TABLE III Challenges and Mitigation

Challenge	Mitigation
False alarms	Score fusion, thresholds tuned on precision
Class imbalance	Class weights, oversampling, focal metrics
Concept drift	Drift monitoring and periodic retraining
Encrypted traffic	Flow metadata and TLS handshake features
Adversarial evasion	Robust training; ensemble diversity

VII. CONCLUSION AND FUTURE WORK

Machine learning has become an essential complement to signature-based security. The reviewed studies show that ensemble classifiers such as Random Forest achieve high detection rates on modern benchmarks and run fast enough for real-time use, while unsupervised detectors add coverage for unknown attacks. Operational success, however, depends on realistic datasets, attention to class imbalance and drift, explainable alerts and close integration with response processes. A hybrid, layered architecture that combines these elements can provide faster, more accurate and more trustworthy threat detection.

1. Acknowledgment

The authors express their sincere gratitude to the Department of Artificial Intelligence and Data Science, Jaihind College of Engineering, Pune, for providing the facilities, guidance and encouragement needed to carry out this work. The authors also thank the faculty members and colleagues whose valuable suggestions helped improve the quality of this review.

REFERENCES

1. D. E. Denning, “An intrusion-detection model,” IEEE Trans. Softw. Eng., vol. SE-13, no. 2, pp. 222–232, 1987.
2. A. L. Buczak and E. Guven, “A survey of data mining and machine learning methods for cyber security intrusion detection,” IEEE Commun. Surveys Tuts., vol. 18, no. 2, pp. 1153–1176, 2016.
3. M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set,” in Proc. IEEE Symp. Comput. Intell. Security Defense Appl. (CISDA), 2009.
4. N. Moustafa and J. Slay, “UNSW-NB15: A comprehensive data set for network intrusion detection systems,” in Proc. Military Commun. Inf. Syst. Conf. (MilCIS), 2015.
5. I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization,” in Proc. 4th Int. Conf. Inf. Syst. Security Privacy (ICISSP), 2018, pp. 108–116.
6. R. Sommer and V. Paxson, “Outside the closed world: On using machine learning for network intrusion detection,” in Proc. IEEE Symp. Security Privacy, 2010, pp. 305–316.
7. Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, “Kitsune: An ensemble of autoencoders for online network intrusion detection,” in Proc. Netw. Distrib. Syst. Security Symp. (NDSS), 2018.
8. N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, “A deep learning approach to network intrusion detection,” IEEE Trans. Emerg. Topics Comput. Intell., vol. 2, no. 1, pp. 41–50, 2018.
9. C. Yin, Y. Zhu, J. Fei, and X. He, “A deep learning approach for intrusion detection using recurrent neural networks,” IEEE Access, vol. 5, pp. 21954–21961, 2017.
10. R. Vinayakumar et al., “Deep learning approach for intelligent intrusion detection system,” IEEE Access, vol. 7, pp. 41525–41550, 2019.
11. H. Liu and B. Lang, “Machine learning and deep learning methods for intrusion detection systems: A survey,” Appl. Sci., vol. 9, no. 20, Art. no. 4396, 2019.

© 2026 IJSRET

12. M. Roesch, “Snort: Lightweight intrusion detection for networks,” in Proc. 13th USENIX LISA Conf., 1999, pp. 229–238.
13. L. Breiman, “Random forests,” Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.
14. C. Cortes and V. Vapnik, “Support-vector networks,” Mach. Learn., vol. 20, no. 3, pp. 273–297, 1995.
15. T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., 2016, pp. 785–794.
16. B. E. Strom et al., “MITRE ATT&CK: Design and philosophy,” MITRE Corp., Tech. Rep., 2018.
17. S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in Proc. NeurIPS, vol. 30, 2017.
18. National Institute of Standards and Technology, “The NIST Cybersecurity Framework (CSF) 2.0,” NIST CSWP 29, 2024.

© 2026 IJSRET