International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(20s) 2026 pp. 1643-1651
https://doi.org/10.70917/ijcisim-2026-6081
Article

AI-DRIVEN CYBER THREAT DETECTION: A FRAMEWORK FOR REAL-TIME IDENTIFICATION AND PREVENTION OF NETWORK ATTACKS

Chaya Lakshmi

HR, Lecturer, Department of Computer Science and Applications, Vishvesvaraya Technological University, Karnataka, India, Email: Chayagowda1@gmail.com

Abstract: Modern enterprise and cloud networks and the Internet-of-Things generate large amounts of data, and malicious activity can occur in small bursts, be spread out across networks, be encrypted, or be deliberately disguised as legitimate activity. The authors present a data-science perspective on real-time cyber-threat detection and detection-driven prevention, with a particular focus on policy-controlled prevention that relies on explainability, supervised learning, and anomaly-oriented learning from streaming flows, and on statistical and temporal feature engineering. The study is not meant to be a statement of a new laboratory experiment but rather a synthesis of secondary benchmarks and a framework. It provides the reproducible datasets such as UNSW-NB15, CICIDS2017, and TON_IoT to develop and evaluate models, and summarizes the latest peer-reviewed studies to inspire models. The system is designed to detect the attack quickly while maintaining its interpretability, while also using the slower computer-based inference system for explanation and retraining for analysts. It suggests accuracy, precision, recall, F1-score, ROC-AUC, false-positive rate, detection latency, throughput, calibration, and distribution drift and class imbalance as evaluation metrics. The evidence published to date indicates that the use of tree ensembles, hybrid deep-learning systems, Transformer-based models, few-shot methods, and explainable AI can all help with network intrusion detection, with the need to map high benchmark scores to production use depending on the datasets and protocols. Leakage-safe splitting, cost-sensitive validation, drift monitoring, confidence-aware response, and human oversight are thus emphasized in the paper. The resulting architecture offers a practical blueprint for adaptive, explainable and operationally deployable network defense.
Keywords: Artificial intelligence; cyber threat detection; data science; intrusion detection system; machine learning; network security; real-time analytics; explainable AI; anomaly detection

1. Introduction

A signature matching issue has now become a data intensive inference problem. Modern networks have diverse users, Apps, Cloud workloads, Mobile endpoints, and connected devices. Meanwhile, an opponent can use reconnaissance, credential abuse, denial of service, botnets, exploitation, lateral movement, command and control, and low and slow tactics which might not be recognized compared to a well-known signature. While traditional rules are still relevant, static signatures are not good enough to stand up to new or changed attack patterns. The complementary capability comes from machine learning (ML) and deep learning (DL), which learn statistical regularities on network flow and detect observations deviating from the expected flow.

The goals of a real-time intrusion detection system (IDS) are generally in conflict with each other: predictiveness and efficiency. A model can have great off-line accuracy but be inappropriate if it is too slow to extract features, if it generates too many false alarms to be of use for analysts, or if the models' decisions are not explainable.

Copyright: © 2026 by the authors

This is a tension that is evident in recent research. The use of transformer based models has been explored for cloud intrusion detection as the attention mechanisms can capture the relationships among traffic features, and explainable-AI research has centered the understanding of the outputs of black-box IDS for security personnel. Recent investigations also reveal that well optimized models can stay highly competitive, an important factor in the context of inference latency and resource usage.

The paper suggests an end-to-end AI-based system for detection and prevention of network attacks in real time. It is not a new benchmark claim, but rather a data-science design that connects data acquisition, feature engineering, model selection, explainability, streaming inference, prevention policy, monitoring and retraining. It also differentiates between published benchmark results and benchmark results that must be created if a future implementation were to be done, so that the metric from one study to another is not considered directly comparable.

2. Research Problem and Motivation

Network defenders have an asymmetric challenge, because there are many more benign packets on the network, attacks come in all shapes and forms, and the impact of an intrusion failure can be far more significant than the impact of processing a normal packet. In the meantime, indiscriminate blocking, based on uncertain predictions can disrupt legitimate services. The research challenge thus is to develop a detection pipeline capable of efficiently and rapidly categorising or scoring traffic, maintaining sensitivity to new and infrequent threats, managing false positives and converting predictions into commensurate prevention actions.

Benchmark datasets show opportunity and challenge. UNSW-NB15 has been designed to be a collection of modern normal traffic and 9 attack categories, and the full collection includes CSV data with 49 generated features, and over 2.54 million records. The published training and testing partitions include 175,341 and 82,332 records respectively. CICIDS2017 provides labeled benign traffic and attack traffic and over 80 flow features generated using CICFlowMeter. TON_IoT expands the benchmark environment to include IoT/IIoT telemetry and multiple data sources. While these datasets have enabled reproducible model development, they are all representative of a specific testbed, a specific period of time, a specific labelling process, and a specific feature space. Production networks can therefore have distribution shift not captured by the benchmark-only evaluation.

3. Objectives and Research Questions

The primary goal is to develop a data-science solution to real-time detection and prevention of cyber threats in networked environment using AI technologies. To build a streaming pipeline for collecting and preprocessing network-flows; To merge statistical, temporal and behavioural characteristics; To accommodate both classical and context-sensitive ML/DL models, and to generate calibrated threat scores and interpretable explanations; To map predictions to safe response policies; To evaluate not only predictive metrics, but also the latency, throughput, burden of false-positive responses, robustness and model drift.

RQ1: What is the appropriate data-processing architecture for low latency AI network threat detection?RQ1: What data processing architecture should be used for low latency AI network threat detection? RQ2: What is the appropriate combination of classical ML, deep learning, and anomaly detection with respect to the accuracy, speed, and adaptability? Consider the following questions: RQ3: What additional metrics are needed to evaluate operational usefulness other than accuracy? RQ4: How can explainability and confidence-aware response mitigate the risk of unsafe automated blocking?

4. Literature Review

To overcome the shortcomings of the older intrusion-detection datasets, Moustafa and Slay proposed a new dataset called UNSW-NB15.To overcome the shortcomings of older intrusion detection datasets, Moustafa and Slay proposed a new one which is called UNSW-NB15. Although their work is outdated, it did create a modernized feature space and a combination of both normal and attack traffic to test network anomaly detection. Later, Sharafaldin, Lashkari and Ghorbani proposed CICIDS2017, which focused on realistic profiles, labeled flows and a wide variety

of modern attacks. The TON_IoT was developed by Alsaedi et al. to support IoT and IIoT security research and telemetry-oriented data that can be used for the evaluation of data-driven IDS.

Recent studies have moved beyond classification to attention and hybrid learning and few-shot adaptation. Long et al. presented a Transformer-based Network Intrusion Detection for Cloud Security, which highlights the rising importance of attention mechanisms in capturing complex feature relationships. They proposed a multi-scale Transformer IDS, which shows how multi-scale representation can be combined with deep learning to achieve intrusion detection.They proposed a multi-scale Transformer IDS, which demonstrates the integration of multi-scale representation with deep learning for intrusion detection. Xu et al. presented a few-shot self-attention method that was designed to minimize reliance on a large number of labeled attack samples, and high detection rate was reported on CICIDS2017 and CICIDS2018 with their mentioned experimental procedure.

Meanwhile, new research warns that more complex architecture does not always yield a more efficient operating system. Another study using data in 2025 showed very high performance of the tree-based methods after preprocessing compared to the others (Random Forest, KNN, Gradient Boosting, and Logistic Regression). A 2025 IEEE study that integrated a DNN and a Random Forest achieved a test accuracy of 98.5% and had a typical processing time of ~50 msec. These results align with the notion of a layered architecture with simpler models optimized for computational efficiency and more complex models called when uncertainty or risk demands further computation.

Explainability is now a key design focus. LIME- and SHAP-based explanations have been used to provide analytics with an understanding of the input features for the ML intrusion-detection output. XAI-IDS research also states that, when it comes to AI predictions for security analysts to interpret and act upon, performance is not enough. A false positive that just creates an alert is less severe than a false positive that blocks a legitimate service, particularly in the case of automated prevention.The consequences of a false positive that only generates an alert is less severe than if it was a false positive that prevented a legitimate service, especially in the context of automated prevention.

The literature thus shows that there is a gap from a practical point of view: many papers optimize a classifier, whereas an operation in cyber defense needs an integrated system. Data quality, leakage prevention, latency, calibration, explainability, response policy, drift, feedback and retraining should all be viewed as part of the same lifecycle. The framework presented here aims to fill this systems level gap.

5. Data Sources and Data Science Design

5.1 Benchmark Data

Table 1 Benchmark datasets supporting the proposed framework

Dataset	Primary role	Representative characteristics	Use in the framework
UNSW-NB15	General network IDS	2,540,044 full records; 49 features; nine attack types	Primary structured-flow benchmark
CICIDS2017	Flow-based attack detection	Benign + common attacks; >80 CICFlowMeter features	Cross-dataset validation / attack diversity
TON_IoT	IoT/IIoT security	Network and telemetry-oriented sources	Heterogeneous/IoT robustness assessment

The proposed experimental protocol employs benchmark datasets as independent evidence sources instead of simply concatenating the datasets without any control. Class taxonomies vary, as do feature names and assumptions of traffic generation. Each data set must then be preprocessed using the specific schema for that set, with a common evaluation layer providing a mapping from native labels to higher-level labels like benign, reconnaissance, denial-of-service, exploitation, malware/botnet, and other malicious behavior where feasible.

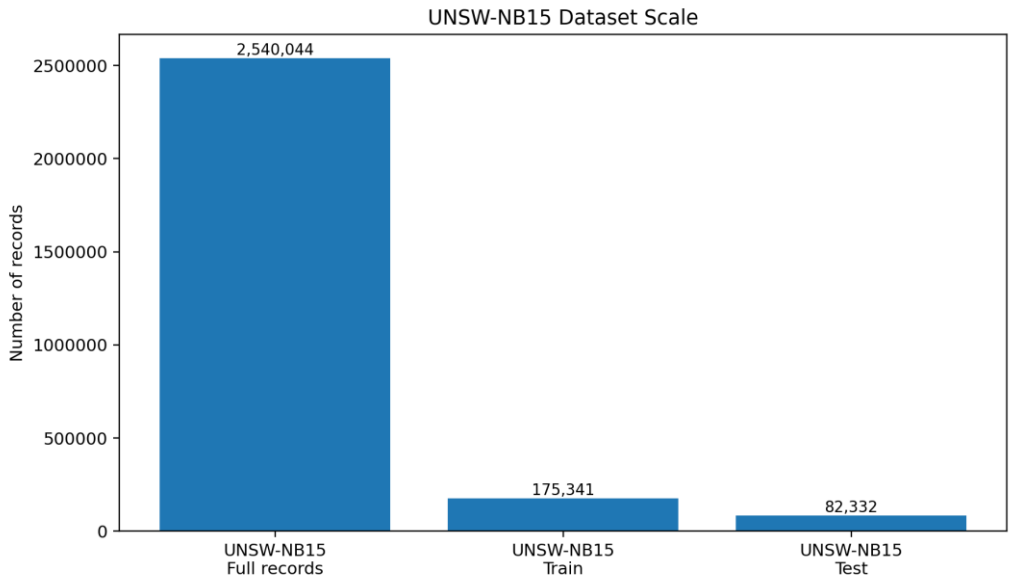


Figure 1 Scale of the official UNSW-NB15 record collections

5.2 Preprocessing and Feature Engineering

Before processing, schema validation, duplicate analysis, handling of missing values, removal of identifiers that may contain labels, categorical encoding, and scaling (if necessary for the used algorithm) are performed. The environment-specific details like IP addresses, precise timings, flow identifiers etc., should be carefully considered since they could enable the model to learn the setup of the lab rather than attack behavior. The partitions of the datasets need to be created prior to any learned preprocessing step to prevent information from the test set to affect the training pipeline.

The feature engineering process is broken down into four groups. The statistical features include count, bytes, duration, rates, inter-arrival times and directional asymmetry. Protocol features encode transport and application behaviour. Temporal features are summarizations of short windows of time, burstiness, repeated connection attempts, and changes in destination diversity. Behavioral features represent the differences from an host, a service and/or a subnet baseline. Implementing feature selection may include methods of mutual information, tree-based feature importance, regularization and stability analysis, but the final set of features should be reviewed to check latency as well as predictive contribution.

5.3 Model Layer

The model layer is purposely heterogeneous. Random Forest and gradient-boosted trees are good fast-path baselines since they can deal with nonlinear interactions, mixed feature importance, and they can infer relatively efficiently. If there is enough labeled data, a small DNN can be used to model high-order interactions. For cases where the temporal or cross-feature dependency is warranted, transformer or attention-based models can be used. A dissimilar behavior signal can be obtained using an anomaly detector such as one class learning, an autoencoder or an Isolation Forest.

The model doesn't make an irreversible decision, it combines several signals into a threat score. Use p_c for the calibrated probability of a malicious class of the supervised classifier, a for a normalized anomaly score, and r for a context risk prior. A simple fusion score is given by $S = w1\,p_c + w2\,a + w3\,r$, where the weights w1, w2 and w3 are all non-negative and the sum of these weights is equal to one. The weights and decision thresholds should be determined on validation data and not only for maximising accuracy as this is considered to be an operational cost.

6. Proposed Real-Time Framework

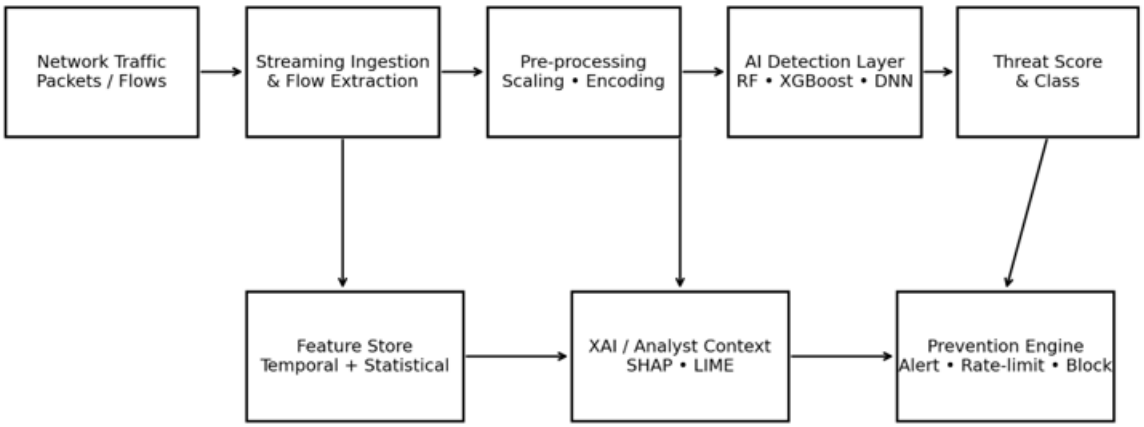


Figure 2 Proposed AI-driven real-time cyber-threat detection and prevention architecture

The framework splits data plane from decision and governance plane. The data plane includes the collection and transformation of packet or flow metadata to streaming features. The decision plane consists of a fast classifier which generates a threat probability and complementary evidence from an anomaly detector and contextual rules. How the prediction is used is driven by explanations, confidence thresholds, prevention policies, audit logs, and analyst feedback in the governance plane.

This separation enables low latency operation. Feature extraction and inference can be done at the end of every flow or at the end of micro-batches, and SHAP/LIME explanations can be computed asynchronously for high risk or sampled events. A high-confidence known attack will result in an immediate, but temporary, action like rate-limiting, connection reset, temporary quarantine or a short-lived firewall rule that are reversible. Normally these events should be enriched and queued for the analysts for review instead of blocking automatically for medium confidence events. Low confidence observations can be kept for drift analysis and re-training.

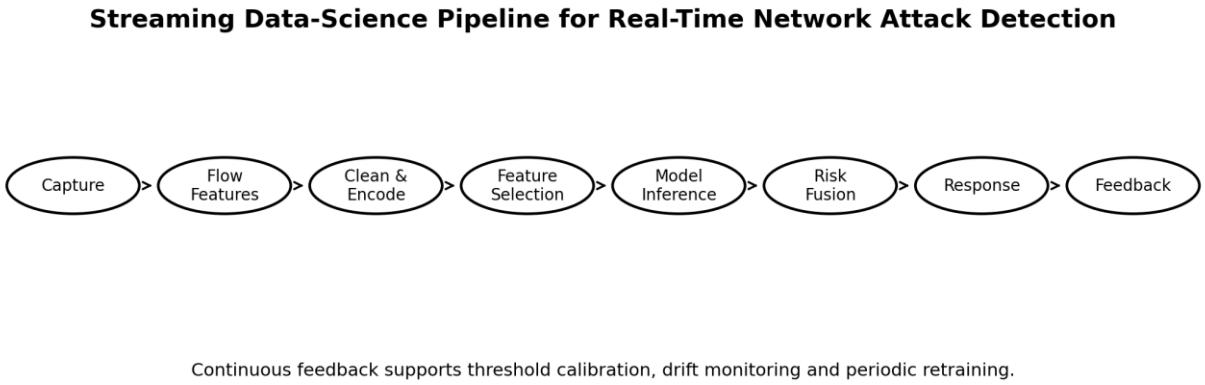


Figure 3 Streaming data-science pipeline for detection, response, and feedback

7. Evaluation Methodology

Leakage-safe train/validation/test partitions should be used in a rigorous evaluation. If the timestamps are known, chronological validation is better, as it better simulates future deployment to a traffic stream. Cross-dataset

testing should be regarded as a robustness test and not a requirement that there is a one-to-one correspondence between all feature schemas. Class imbalance can be overcome by using class weights, optimizing the threshold value, or resampling the data, but only from the training partition, and carefully validate that resampling is not necessary.

Table 2 Evaluation metrics for an operational AI-driven IDS

Metric	Definition / interpretation	Operational relevance
Accuracy	(TP+TN)/(TP+TN+FP+FN)	Useful summary but can be misleading under imbalance
Precision	TP/(TP+FP)	Controls analyst burden and unnecessary blocking
Recall / TPR	TP/(TP+FN)	Measures ability to detect attacks
F1-score	2PR/(P+R)	Balances precision and recall
False-positive rate	FP/(FP+TN)	Critical for production usability
ROC-AUC / PR-AUC	Threshold-independent ranking quality	Useful for threshold selection; PR-AUC is informative for rare attacks
Detection latency	Capture-to-decision time	Determines real-time feasibility
Throughput	Flows or events processed per second	Measures scalability
Calibration	Agreement between confidence and observed frequency	Supports risk-aware automation
Drift robustness	Performance under temporal/domain shift	Indicates durability after deployment

8. Secondary Data Analysis and Graphical Synthesis

In order to provide a data-oriented description without creating a new experiment this paper combines selected performance numbers from the recent peer reviewed literature. The values shown below should not be considered a leaderboard as there are differences in the underlying datasets, class definitions, preprocessing, train-test splits and model objectives. They are intended to demonstrate current performance levels achieved by IDS research and to encourage careful and protocolized replication.

Table 3 Selected published results used for secondary synthesis

Study / model	Reported metric	Reported value	Important caveat
IEEE 2025 DNN + Random Forest	Accuracy	98.5%	Study-specific test set; average processing time reported as 50 ms
IEEE 2025 behavioral ensemble	Accuracy	98.8%	Evaluated on CICIDS2017 under the authors' protocol
Xu et al. 2025 few-shot self-attention	Detection rate	99.90%	CICIDS2017; few-shot protocol, not a conventional full-data comparison
Franklin Open 2025 Decision Tree	Accuracy	99.97%	Optimized preprocessing; study-specific protocol
Franklin Open 2025 Random Forest	Accuracy	99.96%	Optimized preprocessing; study-specific protocol

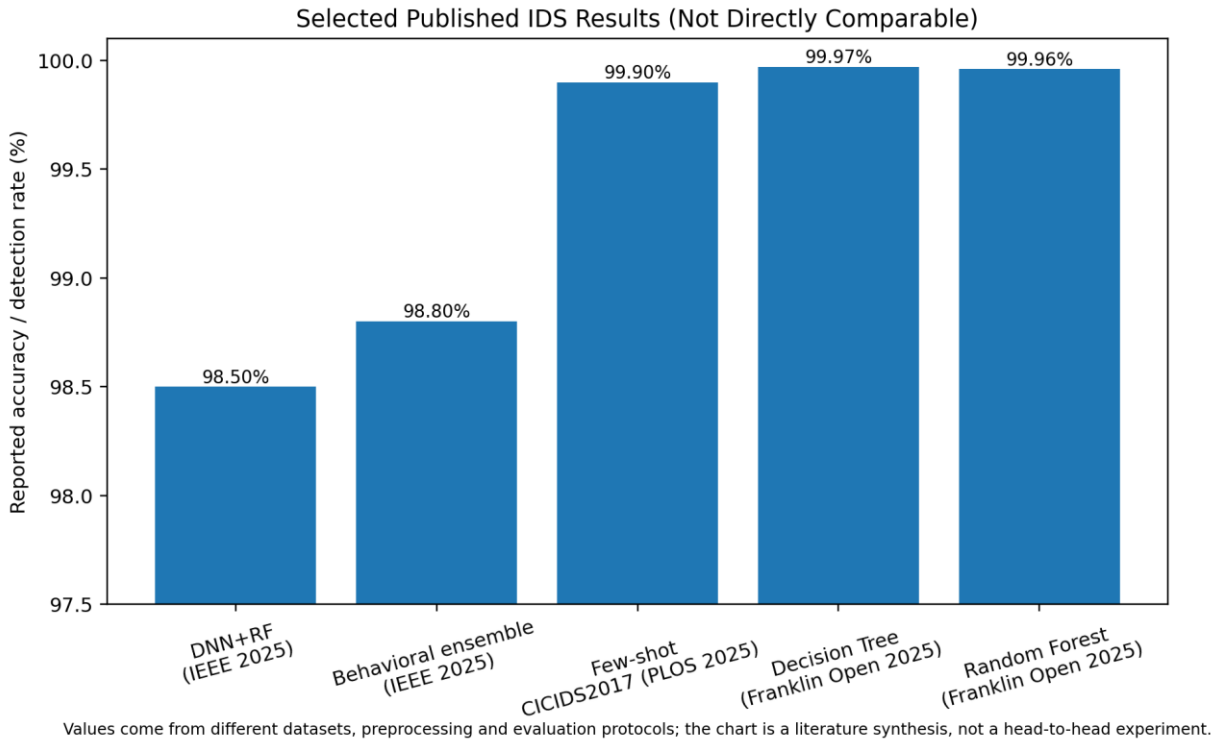


Figure 4 Selected published IDS performance values; studies are not directly comparable

The synthesis suggests three conclusions. First, there is performance on widely used benchmark sets that can be reached both with classical models and deep models. Second, even complex trees may not be better: optimized tree models can be competitive. Third, the operational differences in very-high benchmark scores are not statistically significant in the small.Third, small differences in very high benchmark scores do not matter very much operationally speaking. Confidence intervals and metrics per class should therefore be reported in a production study, not just overall accuracy.

9. Prevention and Response Strategy

Prevention should be reversible and confidence aware. The structure has three levels of responses. Tier 1 generates an alert, and adds host, service, historical and model-explanation context. Tier 2 means temporary containment measures, such as rate limiting, session ending or brief quarantine if the predicted risk is high, but uncertainty remains. Tier 3 permits higher level blocking only under conditions of confidence, corroborating evidence and policy requirements. Stringent human-approval requirements can be applied to critical assets even at high model confidence.

The response engine also needs to prevent any feedback loops. If automated blocking causes a traffic shift, subsequent training data may be skewed by the allowed traffic and not include enough of the blocked traffic. Logging should therefore ensure that the original model score, features, action, disposition by the analyst and the ultimate outcome is preserved. This audit trail can be used in threshold tuning, incident review and unbiased retraining.

10. Explainability, Trust, and Security of the AI Pipeline

There are two roles of explainability.There are two roles of explainability. Global explanations enable engineers to understand the features that the model is built upon, to find leaks, and to detect unstable proxies. Local explanations are used to aid analysts in understanding the reason for a particular flow being flagged. SHAP can offer feature-

attribution values and LIME can approximate local decision behaviour. Explanations are not meant to be statements of cause, but statements of support for decision-making.

The whole AI pipeline is a security target. Attackers might try to use evasion, poisoning, feature manipulation, or model-serving infrastructure exploitation. Defensive controls should include signed model artifacts, access control of training data, immutability of experiment metadata, validation of range of model features, rate limits, confidence drift monitoring, and rollback to a known good model. Retraining should not be done automatically based on analyst labels, but on quality checks instead.

11. Discussion

The proposed framework is based on a data-science lifecycle that includes intrusion detection. This is crucial for network security data, as they are non-stationary in nature. Normal traffic is transformed by new applications, tactics are transformed by attackers, and features are transformed by network architecture. A static model can therefore fall short when the accuracy of the original benchmark is very good. Drift monitoring should measure distribution of features, prediction rates, confidence, per-class recall, false-positive rates, and analyst-identified results.

The literature also indicates that the most suitable practical architecture is probably hybrid. The classical ensembles are fast and yield high tabular performance; deep and attention-based models can capture richer dependencies; anomaly scores can reveal behavior that is not part of the known classes; and XAI can enable analysts to interrogate decisions. The focus point of design is not which single algorithm is best across the board, but how to distribute computation and human attention based on risk.

Reproducibility is a key component of data-science research. All experiments should capture: dataset version, selected columns, preprocessing code, split strategy, random seed, class distribution, hyperparameters, threshold, hardware, inference batch size, and software versions. These statistics are essential for the meaningful comparison and reproduction of a reported accuracy.

12. Limitations

This is a framework study and secondary synthesis paper and cannot be used as justification for the proposed architecture to be newly deployed on a live network or for the literature derived values to be a single controlled experiment. Different preprocessing and validation protocols can influence the cross-paper metrics while benchmark datasets may include artifacts that may not be representative of production traffic. Encryption of traffic reduces payload inspection and adds to the need for flow metadata. Rare attack classes are still challenging to model. Explainability methods may also be misleading or unstable when used without the context of the domain. These constraints are also the criteria for a future implementation and controlled evaluation.

13. Conclusion

AI-based cyber threat detection is best viewed as a problem in data science and security engineering and is not a classifier selection problem. The framework proposed in this paper integrates an acquisition process that is streaming, leakage-aware preprocessing, statistical and behavioral feature engineering, efficient supervised models, anomaly scoring, explainability, calibrated risk fusion, tiered prevention, and continuous monitoring. Recent studies show that tree ensembles, hybrid deep models, Transformers and few-shot methods show high performance levels on benchmark datasets, but production readiness requires false-positive control, latency, robustness, interpretability, and governance. This should therefore entail optimization of the complete decision pipeline and safeguarding of a human decision maker for critical actions in a well-designed real-time IDS. Future empirical testing should be done on UNSW-NB15, CICIDS2017, TON_IoT, and organization's specific traffic, following chronological and cross-domain validation, and fully reporting latency, throughput, calibration, drift, and per-class performance.

REFERENCES

1. Moustafa, N., & Slay, J. (2015). UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). 2015 Military Communications and Information Systems Conference (MilCIS), 1–6. https://doi.org/10.1109/MilCIS.2015.7348942 https://doi.org/10.1109/MilCIS.2015.7348942
2. Moustafa, N., & Slay, J. (2016). The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. Information Security Journal: A Global Perspective, 25(1–3), 18–31. https://doi.org/10.1080/19393555.2015.1125974 https://doi.org/10.1080/19393555.2015.1125974
3. Sharafaldin, I., Habibi Lashkari, A., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP). https://www.unb.ca/cic/datasets/ids-2017.html
4. Alsaedi, A., Moustafa, N., Tari, Z., Mahmood, A., & Anwar, A. (2020). TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems. IEEE Access, 8, 165130–165150. https://doi.org/10.1109/ACCESS.2020.3022862 https://doi.org/10.1109/ACCESS.2020.3022862
5. Moustafa, N., Keshk, M., Debie, E., & Janicke, H. (2020). Federated TON_IoT Windows datasets for evaluating AI-based security applications. 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications, 848–855. https://doi.org/10.1109/TrustCom50675.2020.00114 https://doi.org/10.1109/TrustCom50675.2020.00114
6. Long, Z., Yan, H., Shen, G., Zhang, X., He, H., & Cheng, L. (2024). A Transformer-based network intrusion detection approach for cloud security. Journal of Cloud Computing, 13, 5. https://doi.org/10.1186/s13677-023-00574-9 https://doi.org/10.1186/s13677-023-00574-9
7. Xi, C., Wang, H., & Wang, X. (2024). A novel multi-scale network intrusion detection model with transformer. Scientific Reports, 14, 23239. https://doi.org/10.1038/s41598-024-74214-w https://doi.org/10.1038/s41598-024-74214-w
8. Explainable AI for Intrusion Detection Systems: LIME and SHAP applicability on Multi-Layer Perceptron. (2024). IEEE Access, 12, 30164–30175. https://doi.org/10.1109/ACCESS.2024.3368377 https://doi.org/10.1109/ACCESS.2024.3368377
9. XAI-IDS: Toward proposing an explainable artificial intelligence framework for enhancing network intrusion detection systems. (2024). Applied Sciences, 14(10), 4170. https://doi.org/10.3390/app14104170 https://doi.org/10.3390/app14104170
10. Consensus hybrid ensemble machine learning for intrusion detection with explainable AI. (2024). Journal of Network and Computer Applications. https://doi.org/10.1016/j.jnca.2024.104091 https://doi.org/10.1016/j.jnca.2024.104091
11. Xu, C., Zhan, Y., Chen, G., Wang, Z., Liu, S., & Hu, W. (2025). Elevated few-shot network intrusion detection via self-attention mechanisms and iterative refinement. PLOS ONE, 20(1), e0317713. https://doi.org/10.1371/journal.pone.0317713 https://doi.org/10.1371/journal.pone.0317713
12. Xu, C., Zhan, Y., Wang, Z., et al. (2025). Multimodal fusion based few-shot network intrusion detection system. Scientific Reports, 15, 21986. https://doi.org/10.1038/s41598-025-05217-4 https://doi.org/10.1038/s41598-025-05217-4
13. Deep Learning vs. Machine Learning for Intrusion Detection in Computer Networks: A Comparative Study. (2025). Applied Sciences, 15(4), 1903. https://doi.org/10.3390/app15041903 https://doi.org/10.3390/app15041903
14. Data-driven network intrusion detection using optimized machine learning algorithms. (2025). Franklin Open, 12, 100339. https://doi.org/10.1016/j.fraope.2025.100339 https://doi.org/10.1016/j.fraope.2025.100339
15. Advanced AI-Powered Intrusion Detection Systems in Cybersecurity Protocols for Network Protection. (2025). Procedia Computer Science, 259, 140–149. https://doi.org/10.1016/j.procs.2025.03.315 https://doi.org/10.1016/j.procs.2025.03.315
16. Design and Implementation of Network Intrusion Detection System based on Machine Learning. (2025). 2025 International Conference on Intelligent Systems and Computational Networks (ICISCN). https://doi.org/10.1109/ICISCN64258.2025.10934502 https://doi.org/10.1109/ICISCN64258.2025.10934502
17. Enhancement of Network Security through Machine Learning and Deep Learning Techniques: A Real-Time Intrusion Detection System. (2025). 3rd IEEE International Conference on Industrial Electronics: Developments & Applications. https://doi.org/10.1109/ICIDeA64800.2025.10963052 https://doi.org/10.1109/ICIDeA64800.2025.10963052
18. Intelligent Intrusion Detection System Combining Behavioral Analysis and Machine Learning Methods. (2025). 3rd International Conference on Data Science and Network Security. https://doi.org/10.1109/ICDSNS65743.2025.11168668 https://doi.org/10.1109/ICDSNS65743.2025.11168668
19. Hozouri, A., Mirzaei, A., & Effatparvar, M. (2025). A comprehensive survey on intrusion detection systems with advances in machine learning, deep learning and emerging cybersecurity challenges. Discover Artificial Intelligence, 5, 314. https://doi.org/10.1007/s44163-025-00578-1 https://doi.org/10.1007/s44163-025-00578-1
20. A deep learning/machine learning approach for anomaly based network intrusion detection. (2025). Frontiers in Artificial Intelligence, 8, 1625891. https://doi.org/10.3389/frai.2025.1625891 https://doi.org/10.3389/frai.2025.1625891