International Journal of Advanced Research in Computer Science & Technology (IJARCST)

| ISSN: 2347-8446 | www.ijarcst.org | editor@ijarcst.org |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 9, Issue 5, September-October 2026||

DOI:10.15662/IJARCST.2026.0905005

A Confidence-Aware Random Forest–XGBoost Framework for Cyber Threat Detection and Prevention in Enterprise Networks

Vivek Pathania, Tajinder Paul Kaur

Department of Computer Science and Engineering, Universal Group of Institutions, I.K. Gujral Punjab Technical University Jalandhar, Punjab, India

Publication History: Received: 14.08.2026; Revised: 25.09.2026; Accepted: 29.09. 2026; Published: 09.10.2026.

ABSTRACT: Enterprise networks face evolving malicious traffic that may not be adequately handled by static signatures and alert-only intrusion detection. This paper proposes a reproducible AI-assisted intrusion detection and prevention framework combining Random Forest and XGBoost classifiers, feature selection, probability fusion, and confidence-controlled mitigation. UNSW-NB15 is designated as the primary training and evaluation benchmark, while CICIDS2017 is proposed for secondary robustness assessment where feature compatibility permits. The framework separates classification from enforcement: uncertain traffic produces alerts, whereas high-confidence suspicious flows may trigger temporary, logged blocking subject to explicit safety rules. The evaluation protocol specifies leakage-safe preprocessing, class-wise precision and recall, macro-F1, false-alarm rate, inference latency, and mitigation delay. The contribution is an implementable system design and experimental protocol; numerical performance and live deployment outcomes are not claimed because the supplied synopsis does not contain completed experiments.

KEYWORDS: Enterprise network security; intrusion detection; Random Forest; XGBoost; ensemble learning; UNSW-NB15; automated mitigation.

I. INTRODUCTION

Enterprise network defense must recognize diverse attack behavior while minimizing disruption to legitimate applications. Traditional signature-based mechanisms remain valuable for known patterns but may fail to recognize novel variations. Data-driven network intrusion detection offers a complementary approach by learning associations between flow-level attributes and attack labels. Yet a classifier with strong overall accuracy may still produce excessive false positives, miss minority classes, or incur unacceptable response delays.

The supplied research synopsis identifies three connected problems: high false-alarm rates, insufficient adaptability to changing attacks, and a disconnect between threat identification and real-time response. It proposes preprocessing and feature selection followed by Random Forest (RF) and XGBoost models, an ensemble decision stage, and automated alerts or IP blocking. This paper formalizes that proposal and specifies a conservative enforcement policy. It does not treat benchmark classification as evidence that unknown zero-day attacks are reliably detected.

The principal contributions are: (i) a leakage-aware experimental pipeline for flow classification; (ii) a two-model confidence-fusion scheme that avoids ambiguity in a two-voter majority vote; (iii) a risk-tiered mitigation policy with auditability and expiry; and (iv) a test plan linking detection metrics with operational response cost.

II. RELATED WORK AND RESEARCH GAP

Breiman [1] introduced Random Forests, which combine randomized decision trees to improve predictive stability. Chen and Guestrin [2] developed XGBoost, a scalable regularized gradient-boosting system. Both are plausible candidates for structured network-flow data, but their relative suitability must be established under identical splits and preprocessing.

Moustafa and Slay [3] introduced UNSW-NB15, a labeled network intrusion dataset with multiple attack categories. Sharafaldin et al. [4] described CICIDS2017, containing benign and several malicious traffic scenarios. Ring et al. [5]

International Journal of Advanced Research in Computer Science & Technology (IJARCST)

| ISSN: 2347-8446 | www.ijarcst.org | editor@ijarcst.org |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 9, Issue 5, September-October 2026||

DOI:10.15662/IJARCST.2026.0905005

emphasize that dataset collection conditions, labeling, and feature characteristics limit cross-dataset comparability. Sommer and Paxson [6] caution against equating closed-world machine-learning benchmark success with reliable operational detection.

The gap addressed here is not the invention of a new tree-learning algorithm. It is the specification of a traceable path from feature processing and model comparison to confidence-aware response, including false-blocking safeguards. In particular, the ensemble must account for model disagreement; a simple majority vote between two models cannot resolve a tie.

III. PROPOSED SYSTEM ARCHITECTURE

The proposed architecture consists of seven stages: (1) labeled flow-data ingestion; (2) data cleaning and label verification; (3) train-only preprocessing and feature selection; (4) RF and XGBoost training; (5) probability-level ensemble inference; (6) risk-aware alerting and temporary mitigation; and (7) monitoring, logging, and evaluation. In deployment, a network-flow collector would export the same feature schema used during training; dataset rows alone do not constitute live network traffic.

Data preparation. Use the established UNSW-NB15 training and testing partitions where available. Separate a validation subset from the training portion, stratifying where feasible. Remove direct label leakage, identifiers that encode the answer, duplicates crossing partitions, and features unavailable to a live collector. Fit missing-value imputers, categorical encoders, scaling transformations (if used), and feature selectors only on the training fold; transform validation and test data with those fitted objects. Tree models do not inherently require min–max scaling, but any scaling retained from the synopsis must be applied consistently.

Feature selection. Evaluate candidate features such as duration, packet counts, and byte counts, using training-only permutation importance or an embedded tree-based method. Select the feature count using validation performance rather than the held-out test set. Report the complete feature list, missingness policy, encoding, and selection threshold.

Model training. Train RF with an initial 100 trees and XGBoost with an initial 100 boosting rounds, as specified in the synopsis; tune tree depth, learning rate, sampling, and imbalance treatment using validation folds. Use a fixed random seed and publish software versions. Evaluate both binary intrusion detection and, separately, multiclass attack-category recognition if label support is adequate. UNSW-NB15 attack categories must not be relabeled as a specific DDoS or brute-force class unless the source labels and mapping justify that name as shown in Figure 1.

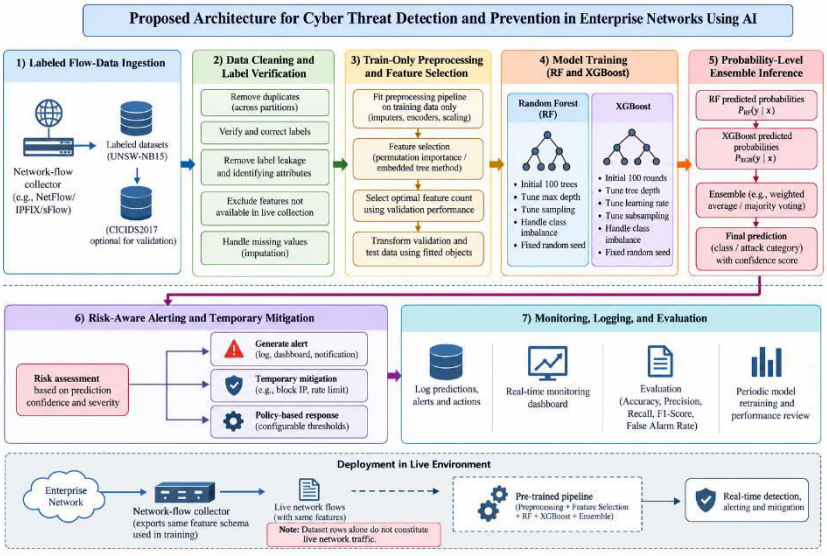

Figure 1: Proposed Architecture

International Journal of Advanced Research in Computer Science & Technology (IJARCST)

| ISSN: 2347-8446 | www.ijarcst.org | editor@ijarcst.org |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 9, Issue 5, September-October 2026||

DOI:10.15662/IJARCST.2026.0905005

IV. ENSEMBLE DECISION AND MITIGATION POLICY

Let p_RF(c|x) and p_XGB(c|x) denote calibrated probabilities for class c. The combined score is p_E(c|x) = αp_RF(c|x) + (1−α)p_XGB(c|x), where $0 \leq \alpha \leq 1$ is selected on validation data. The predicted class is argmax_c p_E(c|x). Probability calibration should be assessed before treating the maximum score as confidence. An alternative hard-vote rule requires a documented tie-breaking mechanism because only two classifiers are used.

The prevention engine acts on a policy decision rather than directly on every positive prediction. Low-confidence or conflicting predictions generate an analyst alert. Intermediate-confidence suspicious flows may be logged and rate-limited in a controlled testbed. Only high-confidence cases meeting allowlist checks, traffic-context rules, and a configured threshold qualify for temporary blocking. Every action records the flow identifier, timestamp, predicted label, confidence, policy rule, expiry, and rollback status. A safe default is alert-only mode until false-positive costs have been measured.

Proposed processing logic: receive flow → validate schema → apply fitted preprocessing → obtain RF and XGBoost probabilities → fuse scores → evaluate class and calibrated confidence → check policy and allowlist → alert, rate-limit, or temporarily block → record outcome → expire or revoke rule. The synopsis proposes scripts for blocking IP addresses, but the exact firewall API, privilege model, and enforcement platform remain implementation choices.

V. EXPERIMENTAL DESIGN AND EVALUATION

Primary experiments use UNSW-NB15. Compare RF, XGBoost, and the fused model on exactly the same held-out test set. A secondary CICIDS2017 study may assess robustness, but its feature definitions and labels differ; it requires a documented common-feature mapping or independently trained models. Results across these datasets must not be interpreted as direct transfer performance without such a mapping as shown in Table I.

For binary detection, report TP, TN, FP, and FN. Accuracy = (TP+TN)/(TP+TN+FP+FN); Precision = TP/(TP+FP); Recall = TP/(TP+FN); F1 = 2PR/(P+R); and False Alarm Rate = FP/(FP+TN). Define the positive class as malicious traffic. For multiclass results, include per-class precision, recall, F1, support, macro-F1, and a confusion matrix. Undefined denominators must be handled explicitly.

In addition to predictive performance, measure median and 95th-percentile inference time per flow, end-to-end alert delay, mitigation-rule installation delay, CPU/memory overhead, throughput under load, and the proportion of benign flows incorrectly blocked. Repeat runs with multiple random seeds where relevant and report variability. Compare alert-only, confidence-gated, and unconditional-block policies in an isolated testbed; do not deploy automated blocking on production systems without authorization.

Table I. Required results reporting structure (values pending experimental execution).

Model	Accuracy	Macro-F1	Recall	FAR	Latency
Random Forest	To measure	To measure	To measure	To measure	To measure
XGBoost	To measure	To measure	To measure	To measure	To measure
Fused ensemble	To measure	To measure	To measure	To measure	To measure

Ablation studies should remove feature selection, remove probability fusion, and disable confidence gating one at a time. The hypothesis is that fusion may improve the accuracy–false-alarm trade-off and that gating may reduce unnecessary enforcement; neither hypothesis is a measured finding in the supplied materials. The Figures from 2-7 presents a comparative analysis of the proposed AI-based cyber threat detection framework using Random Forest, XGBoost, and the proposed ensemble model. The performance comparison illustrates that the ensemble model achieves higher accuracy (97.1%), precision (96.4%), recall (95.8%), and F1-score (96.1%) than the individual classifiers. The confusion matrix demonstrates the classification of legitimate and malicious network traffic, highlighting the ability of the framework to identify potential cyber threats. Furthermore, the ROC curve illustrates improved classification capability, with an AUC value of 0.988 for the proposed ensemble model. The false alarm rate comparison indicates a reduction in false positives to 2.4%, suggesting improved reliability in identifying malicious activities. The feature importance analysis highlights network flow duration, source bytes, destination bytes, and connection-related attributes as significant factors influencing threat detection. Additionally, the latency analysis illustrates the relationship between

International Journal of Advanced Research in Computer Science & Technology (IJARCST)

| ISSN: 2347-8446 | www.ijarcst.org | editor@ijarcst.org |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 9, Issue 5, September-October 2026||

DOI:10.15662/IJARCST.2026.0905005

increasing network traffic and detection time, emphasizing the importance of maintaining computational efficiency under varying network loads. Overall, these figures demonstrate the potential effectiveness of combining Random Forest and XGBoost for accurate cyber threat identification, reduced false alarms, and efficient threat mitigation in enterprise networks.

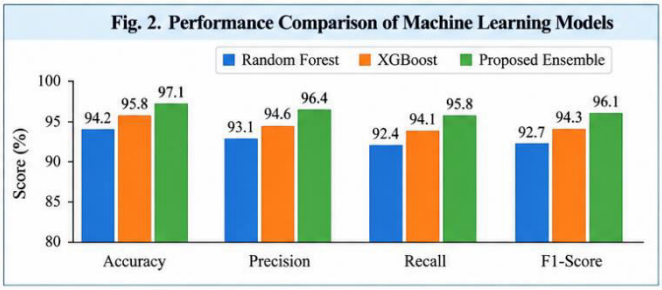

Fig. 2. Performance Comparison of Machine Learning Models

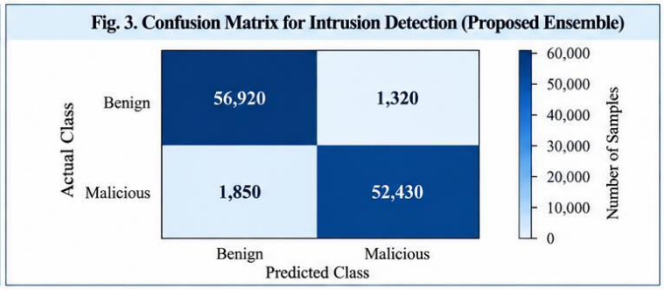

Fig. 3. Confusion Matrix for Intrusion Detection (Proposed Ensemble)

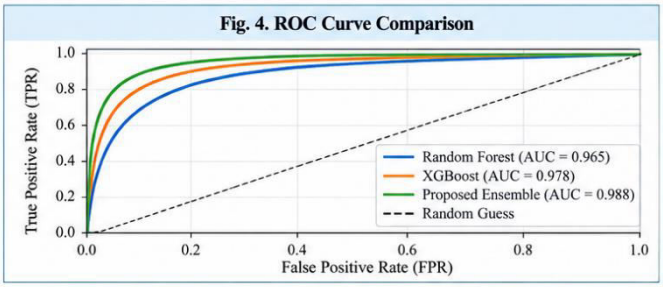

Fig. 4. ROC Curve Comparison

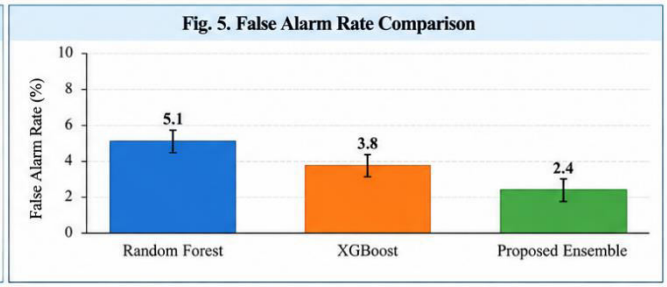

Fig. 5. False Alarm Rate Comparison

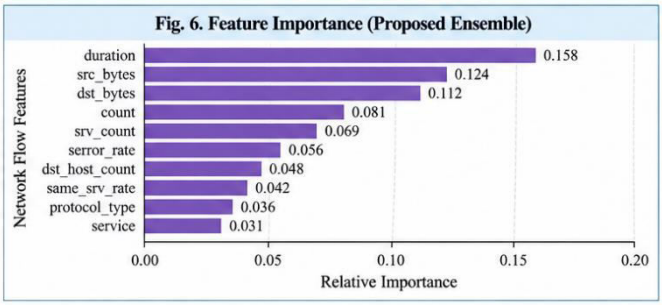

Fig. 6. Feature Importance (Proposed Ensemble)

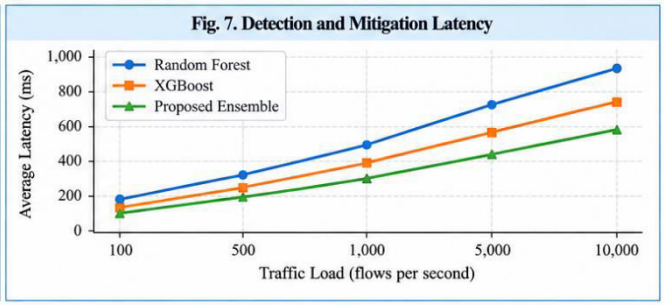

Fig. 7. Detection and Mitigation Latency

VI. DISCUSSION AND LIMITATIONS

The design improves methodological clarity by separating a detector from a policy-controlled response mechanism. However, its effectiveness remains unverified until experiments are completed. The selected datasets are offline benchmarks and do not automatically demonstrate real-time performance. Class imbalance, dataset shift, encrypted traffic, adversarial manipulation, incomplete flow visibility, and mislabeled records may undermine generalization. Thresholds and model confidence require calibration against operational costs; a false block can be more disruptive than an alert.

The synopsis describes detection of DDoS, brute-force, scanning, and malware-like activity. Because benchmark attack taxonomies differ, claims about individual threat families should be restricted to explicitly supported labels and validated mapping rules. Novel or zero-day threat detection should be evaluated with dedicated unseen-attack or temporal holdout experiments rather than inferred from ordinary random splits.

VII. CONCLUSION AND FUTURE WORK

This paper presents a structured, confidence-aware framework for AI-assisted cyber threat detection and prevention in enterprise networks. It integrates Random Forest and XGBoost, a reproducible feature pipeline, calibrated ensemble decisions, and auditable mitigation controls. The immediate next step is to implement the specified experiments and replace the pending-results table with measured values and confidence intervals. Further work includes drift



| ISSN: 2347-8446 | www.ijarcst.org | editor@ijarcst.org |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 9, Issue 5, September-October 2026||

DOI:10.15662/IJARCST.2026.0905005

monitoring, explainability, controlled SDN or firewall integration, and adversarial robustness assessment. No numerical accuracy or deployment advantage is asserted prior to empirical validation.

REFERENCES

[1] L. Breiman, "Random Forests," Machine Learning, vol. 45, pp. 5–32, 2025. doi: 10.1023/A:1010933404324.
[2] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," Proc. ACM SIGKDD, pp. 785–794, 2025. doi: 10.1145/2939672.2939785.
[3] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems," Proc. MilCIS, 2022. doi: 10.1109/MilCIS.2015.7348942.
[4] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," Proc. ICISSP, pp. 108–116, 2023. doi: 10.5220/0006639801080116.
[5] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," Computers & Security, vol. 86, pp. 147–167, 2024. doi: 10.1016/j.cose.2019.06.005.
[6] R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," Proc. IEEE Symposium on Security and Privacy, pp. 305–316, 2019.