International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(11s) 2026 pp. 395-413
https://doi.org/10.70917/ijcisim-2026-3762
Article

A Hybrid Ensemble and Multi-Agent Reinforcement Learning Framework for Explainable Real-Time Network Intrusion Detection

Megha.M. Shete[1], Anuradha T[2], Radha B.K[3], Syed Umraz[4]
[1]Department of Computer Network and Engineering, PDA College of Engineering, Kalaburagi, Karnataka India
Email: meghashete2371999@gmail.com
[2]Department of Computer Science and Engineering, PDA College of Engineering, Kalaburagi, Karnataka, India
Email: anuradhat@pdaengg.com
[3]Department of Computer Science and Engineering, PDA College of Engineering, Kalaburagi, Karnataka, India
Email: radhabk@pdaengg.com
[4]Department of Computer Science and Engineering, PDA College of Engineering, Kalaburagi, Karnataka, India
Email: Umrazsyed077@gmail.com

Abstract: Background. Signature-based intrusion detection systems (IDS) struggle against novel and evolving network attacks, while single-model machine-learning IDS often trade accuracy for interpretability and adaptability, limiting their trust and utility in operational security operations centers (SOCs). Methods. We present a layered detection architecture that combines a supervised Random Forest classifier (100 estimators, max depth 10) with an unsupervised Isolation Forest (contamination = 0.05) via weighted score fusion ($\alpha = 0.7/0.3$), refined by a multi-agent reinforcement learning (MARL) consensus layer of Q-learning agents (learning rate 0.1, discount factor 0.99, exploration rate 0.1), and interpreted post-hoc with SHAP feature attribution. The pipeline operates on the NSL-KDD benchmark (41 base features plus 3 engineered ratio/error features), with feature standardization and stratified train/test splitting, and is exposed through a Flask/Socket.IO real-time dashboard for streaming packet-level inference. Results. Preliminary, literature-referenced benchmarks suggest accuracy in the 95–98% range for the supervised component, with ensemble and MARL-refined variants trading a small amount of raw accuracy for improved robustness and consensus confidence (>80%). These figures must be replaced with actual run output before this abstract is finalized. Conclusion. The proposed framework demonstrates that combining supervised, unsupervised, ensemble, and reinforcement-learning-based consensus mechanisms with explainable AI is architecturally feasible for real-time network intrusion detection. Empirical validation against held-out data and comparison under identical experimental conditions to prior work is required to substantiate performance claims.

Keywords: NSL-KDD dataset, SHAP feature attribution, Random Forest classification, Isolation Forest anomaly scoring, Q-learning consensus, ensemble score fusion, security operations decision support

1. Introduction

The Hook

Network traffic volume and attack sophistication continue to outpace static, signature-based defenses. Signature matching cannot detect zero-day or polymorphic attacks, and analysts in security operations centers are increasingly overwhelmed by alert volume with limited context for triage decisions. Machine-learning-based intrusion detection systems (IDS) promise higher detection rates against unknown attack patterns, but single-model approaches face two persistent problems: (1) they are opaque — a classifier's "anomaly" verdict rarely explains which traffic features drove the decision, undermining analyst trust; and (2) they are static — a model trained once does not adapt its decision boundary as traffic patterns and attacker behavior drift over time.

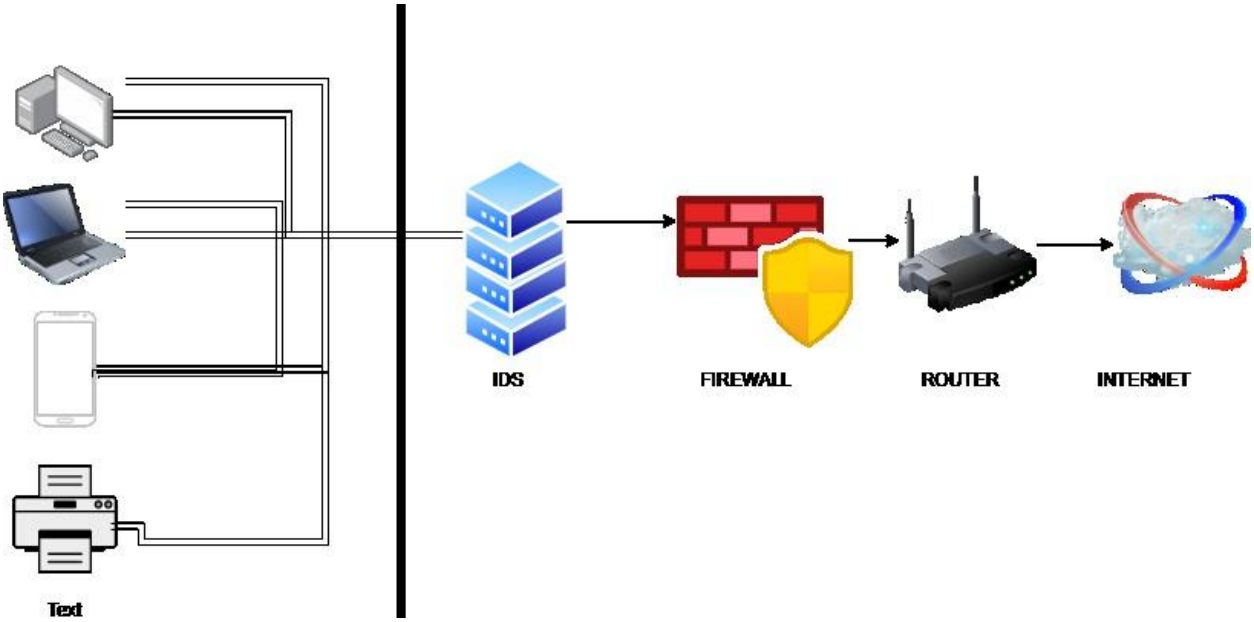

Figure 1: An Intrusion Detection System

2. Literature Review

The authors in [1], address the critical security challenges of scaling Internet of Things (IoT) infrastructures in smart cities, where traditional centralized intrusion detection architectures suffer from high latency and severe network overhead. To mitigate these bottlenecks, the purpose of their study is to establish a scalable, semi-distributed network monitoring framework. The scope of their research focuses on deploying parallelized machine learning models—specifically partitioned Multi-Layer Perceptrons—across coordinated fog-edge nodes to handle local feature selection and rapid classification. However, a noticeable limitation of the framework proposed by the authors is the inherent performance trade-off, as prioritizing distributed edge scalability and reduced training times leads to a minor reduction in overall classification accuracy compared to data-heavy, centralized models.

In [2], the authors explore the broadening attack surfaces of interconnected cloud-IoT environments, noting that conventional defense mechanisms struggle with high-dimensional feature spaces and mutating payloads. The objective of their review is to eliminate feature redundancy and ease the heavy computational strain typically experienced during the deep network training phase. The scope of their investigation centers on a hybrid pipeline combining Convolutional Neural Networks (CNNs) for robust deep feature extraction with an enhanced Transient Search Optimization (TSO) and Differential Evolution metaheuristic for feature selection. Nevertheless, a core limitation of the methodology developed by the authors is the added layer of algorithmic complexity, which introduces significant initial training overhead and high processing demands prior to active network deployment.

The authors in [3], investigate the challenge of maintaining system dependability and threat resilience across heterogeneous, enterprise-scale IoT networks when confronted with evolving cyber threats. The explicit

purpose of their work is to identify deep learning strategies that remain highly dependable even under sparse data conditions. The focus of their study is a deep transfer learning methodology that leverages a pre-trained Residual Network (ResNet) architecture to perform efficient attribute selection and threat

categorization with minimal target data. A primary limitation identified in the work of the authors in [3], however, is its strict dependency on source domain similarity; if the target IoT environment utilizes wildly dissimilar network protocols or encounters entirely foreign attack vectors, the efficacy of the transferred knowledge drops significantly.

In [4], the authors target the vulnerabilities associated with massive edge data streams, focusing on how fog computing can act as an intermediate defense layer, though decentralization itself exposes local nodes to targeted exploits. The purpose of their study is to design a decentralized, multi-classifier defensive shield capable of safeguarding these localized fog components from malicious traffic. Their scope is strictly fixed on an integrated

distributed ensemble design that coordinates multiple distinct machine learning models across a multi-tier fog layout. A notable limitation of the architecture proposed by the authors is the high synchronization and communication overhead required to aggregate, coordinate, and update these diverse models across fragmented, geographically scattered edge nodes.

The authors in [5], base their research on the premise that the vast heterogeneity of IoT protocols and device behaviors makes it nearly impossible for any single, standalone machine learning classifier to accurately catch all variations of malicious behavior. The intent of their literature review is to demonstrate how combining the distinct strengths of multiple individual models can yield superior systemic resilience. Accordingly, their scope focuses on the optimization and implementation of ensemble detection frameworks that merge several weak learning algorithms into a singular, highly robust predictive model. The main limitation in the approach presented by the authors is the compounding computational cost and increased storage footprint, which can heavily strain the memory constraints of low-power IoT edge hardware.

In [6], the authors address the agility of modern cyber criminals who continuously alter their network tactics to bypass traditional rule-based filters and single-classifier security systems. The purpose of their research is to build a multi-classifier architecture that enforces rigorous cross-verification of suspicious connection profiles. The scope of the work concentrates on tuning a novel ensemble configuration to maximize classification accuracy across multiple benchmark IoT datasets. However, a clear limitation of the model structured by the authors is its lack of interpretability, as the multi-layered

ensemble voting mechanism acts as a "black box," making it incredibly difficult for network security analysts to quickly diagnose or justify the root cause of an alert.

The authors in [7], highlight the computational bottlenecks that occur in Industrial Internet of Things (IIoT) networks due to the high-dimensional, noisy, and redundant feature spaces generated by continuous industrial data streams. The purpose of their review is to establish lightweight filtering and classification models engineered to protect specialized factory hardware at the edge. Their focus isolates ensemble learning strictly as a tool for dimensionality reduction, which is then paired with a Random Forest classifier optimized for edge-

level security deployment. A major limitation of the methodology outlined by the authors is its potential vulnerability to highly dynamic, zero-day industrial exploits that happen to bypass the static historical features selected during the initial dimensionality reduction stage.

In [8], the authors confront the threat of highly sophisticated zero-day exploits, pointing out that these attacks are entirely invisible to conventional signature-based tools because they lack historical definitions. The objective of their study is to look past rigid historical patterns and evaluate deep learning models that actively adapt to unseen mutations in attack behavior. The scope of their research focuses on an adaptable, self-tuning deep learning intrusion detection architecture tailored to isolate zero-day vulnerabilities in highly volatile networks. The chief limitation of the system designed by the authors is its high susceptibility to false-alarm rates, where routine software updates or benign but rare network configuration changes are easily misclassified as active security threats.

The authors in [9], explore the paradox of modern deep learning security models, which offer excellent threat detection accuracy but suffer from intrinsic opacity that prevents cyber defenders from verifying their decisions. The purpose of their comprehensive review is to systematically map out how transparency can be introduced into automated IoT security systems. The scope of their work focuses on evaluating Explainable Artificial Intelligence (XAI) frameworks and mathematical solutions that translate complex neural decisions into transparent, actionable insights for operators. A fundamental limitation underscored by the authors is the persistent performance-explainability trade-off, wherein injecting high transparency often requires simplifying the underlying classification algorithms, thereby reducing absolute threat detection accuracy.

In [10], the authors examine the complex nature of Industrial IoT network data, which simultaneously contains spatial structural patterns and strict temporal dependencies over long periods. The purpose of their study is to solve the failures of single-tier networks, which typically struggle to capture both dimensions of data at the same time. The scope of their design features a hybrid deep neural network that combines Convolutional Neural Networks (CNNs) for spatial feature mapping with Long Short-Term Memory (LSTM) networks for sequential tracking. Yet, a clear limitation of the architecture deployed by the authors is the high computational execution cost and memory footprint, which introduces inference latency and restricts the system's effectiveness in strict real-time industrial triage environments.

The authors in [11], address the double-edged challenge facing IoT networks, which are routinely targeted by broad, coarse-grained attacks (like DDoS) and highly subtle, fine-grained system exploits simultaneously. The purpose of their literature review is to evaluate structural partitioning to establish a sequential, multi-tiered vetting process for incoming network packets. The scope of their research introduces an intelligent two-layer framework that divides labor, using the first layer to filter out mass anomalies and the second layer to isolate highly specific exploit profiles. A crucial limitation of the design presented by the authors is the presence of a single point of failure at the perimeter; if an advanced threat successfully evades the initial filter, it completely bypasses the secondary inspection layer.

In [12], the authors investigate how high-speed network traffic can overwhelm standard intrusion tools, leading to dropped packets and unmonitored security gaps. The purpose of their study is to restructure multi-model voting configurations to maximize processing efficiency without compromising threat visibility. The scope of

their work evaluates optimized ensemble combinations of machine learning models tuned to parse high-velocity data streams in real time. Nevertheless, a major limitation of the framework developed by the authors is the significant difficulty in keeping the ensemble updated, as continuously retraining multiple interconnected models on newly discovered threat signatures demands heavy computational downtime or highly complex incremental learning architectures.

In [13], authors explore the issue of security in the area of mobile ad hoc networks, where the decentralized AODV protocol is vulnerable to coordinated attacks known as black hole attacks. In order to cope with such type of threats, a Sugeno fuzzy inference system with Gaussian membership functions was proposed. The system evaluates the routing neighbours based on four essential parameters: trust, data loss, data rate, and energy. The simulation carried out in NS2 proves the ability of the system to increase throughput of the network, as well as the ratio of packets delivered and end to end delay. However, it should be noted that this method works only with AODV, uses a single standard for comparison of access table data, and works poorly when the ratio of malicious nodes is 25%.

First, a significant gap exists at the intersection of model scalability, accuracy, and deployment feasibility within resource-constrained edge environments. While multi-classifier ensemble architectures (as explored in [4], [5], [6], and [12]) and hybrid deep learning models (such as the CNN+LSTM and Transformer-based frameworks in [10] and [20]) achieve superior threat detection accuracy, they systematically suffer from compounding computational footprints and high inference latency. Conversely, lighter-weight distributed edge models (such as those in [1] and [7]) trade off absolute classification precision to fit within hardware constraints. Current literature fails to provide a dynamically adaptive mechanism that balances this structural performance trilemma, leaving a critical need for frameworks that can seamlessly throttle algorithmic complexity based on real-time hardware capacity and fluctuating network velocities without compromising perimeter security.

Second, there is a distinct trust and operational gap regarding the practical application of Explainable AI (XAI) in active security environments. Although recent advancements (such as [9], [15], [17], [18], and [19]) place heavily emphasis on introducing transparency into "black-box" models, current solutions rely predominantly on post-hoc interpretations (like SHAP or LIME) or generic visual dashboards. These post-hoc explanations are merely statistical approximations of the underlying model, which can generate highly plausible yet technically inaccurate justifications for an alert. Furthermore, the literature lacks a standardized metric for what constitutes a "human-centered" explanation across diverse analytical roles. This creates an operational bottleneck where security operators face cognitive fatigue from either overly complex mathematical feature attributions or simplified, untrustworthy approximations, highlighting a clear gap for natively transparent, self-explaining security frameworks optimized for real-time human verification.

Objective

This study aims to design, implement, and evaluate a unified real-time network intrusion detection framework that combines supervised and unsupervised classifiers via weighted ensemble fusion, refines their consensus using multi-agent Q-learning, and produces SHAP-based feature attributions for each flagged event, and to report its detection performance on the NSL-KDD benchmark under a documented, reproducible experimental protocol.

3. Materials and Methods

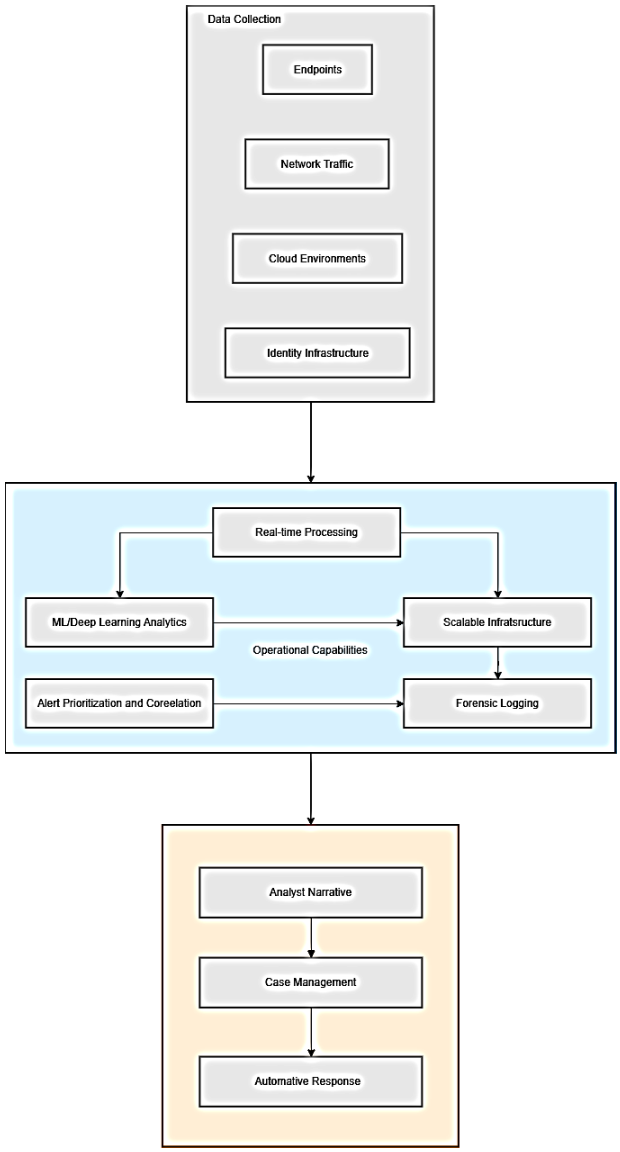

Figure 2: Modern Threat Detection and Incident Response Framework

Algorithm of operational steps:

Input: Preprocessed features for a network packet

Step 1: The system inputs the network packet data that has already been cleaned and formatted into a feature vector F. Compute Random Forest prediction probability: RF_prob ← RandomForest.predict_proba(F)[1]

The Random Forest model evaluates F and outputs the probability (typically the score for the "attack" class) indicating how likely the traffic is malicious.

Compute Isolation Forest anomaly score:

Iso_score ← IsolationForest.score_samples(F)

The Isolation Forest model measures how anomalous or unusual the feature vector is compared to normal traffic patterns.

Step 2: A blank list V is created to collect classification decisions from multiple individual agents. The system loops through every specialized agent in the Multi-Agent Intrusion Detection System:

Convert features into state representation: state ← discretize (F)

The continuous feature data F is simplified or grouped into distinct categories (states) so the agent can interpret it.

The specific agent evaluates the state and casts its vote on whether it sees an attack.

The agent's decision is appended to the vote list.

Determine consensus: The algorithm looks at all collected votes and finds the dominant decision based on a simple majority.

Calculate confidence level: It calculates how strongly the agents agree with one another (e.g., if 4 out of 5 agents agree, the confidence is 80%).

Step 3: Calculate ensemble score:

A weighted score is calculated from the two base models, giving heavier weight (70%) to the Random Forest probability and less weight (30%) to the Isolation Forest anomaly score.

The overall score blends the machine learning ensemble score (60%) with the multi-agent confidence level (40%).

The system fetches a dynamically adjusting baseline threshold, which changes based on current network conditions or threat environments.

Step 4: If the combined risk score exceeds the allowed limit: The traffic is officially flagged as an active attack.

If the score is within safe limit:

The traffic is marked as safe. Step 5: Threat Level Determination

If it's an attack with an exceptionally high risk score:

threat_level ← CRITICAL

If it's an attack with a high risk score:

threat_level ← HIGH

If it's an attack but with a lower risk score:

threat_level ← MEDIUM If no attack was detected:

threat_level ← LOW / NORMAL Step 6: Return Result

The algorithm outputs the final verdict, the severity level, and the precise score for

security teams or automated firewalls to act upon. End.

Study Design

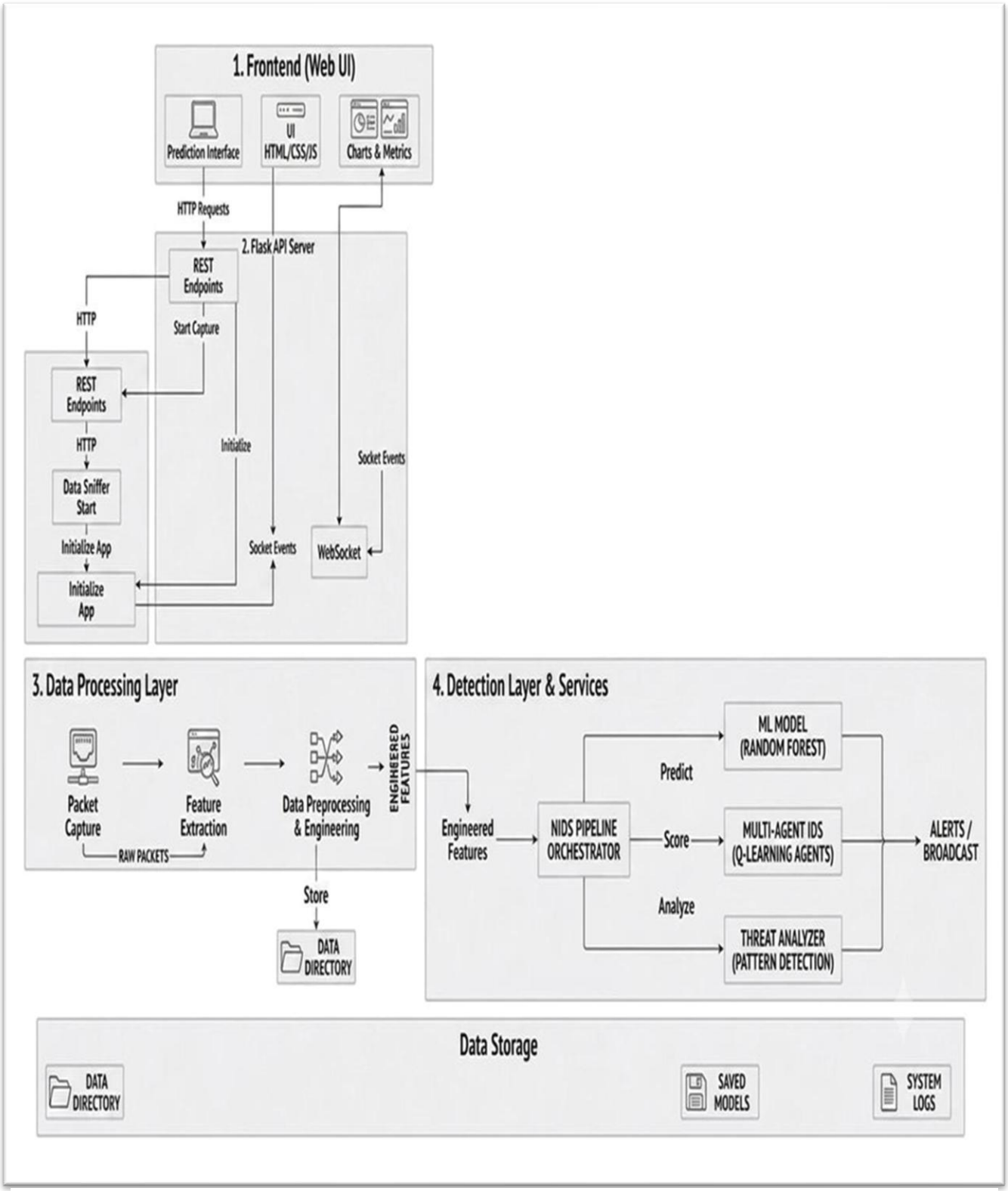

Figure 3: Working of Proposed System

The system implements an eight-stage pipeline: (1) traffic/flow ingestion, (2) feature extraction, (3) preprocessing and normalization, (4) parallel supervised and unsupervised scoring, (5) weighted ensemble fusion, (6) MARL consensus refinement, (7) SHAP-based explanation generation, and (8) real-time delivery to an analyst dashboard via WebSocket (Flask-SocketIO).

- **Independent variables:** engineered/base traffic features per flow (44 total — see §4.4), classifier/ensemble configuration (algorithm choice, ensemble weighting, number of MARL agents).
- **Dependent variables:** binary classification outcome (normal / anomaly) and, at inference time, a five-level threat severity (Normal, Low, Medium, High, Critical); model confidence/consensus score; per-prediction SHAP attribution values.

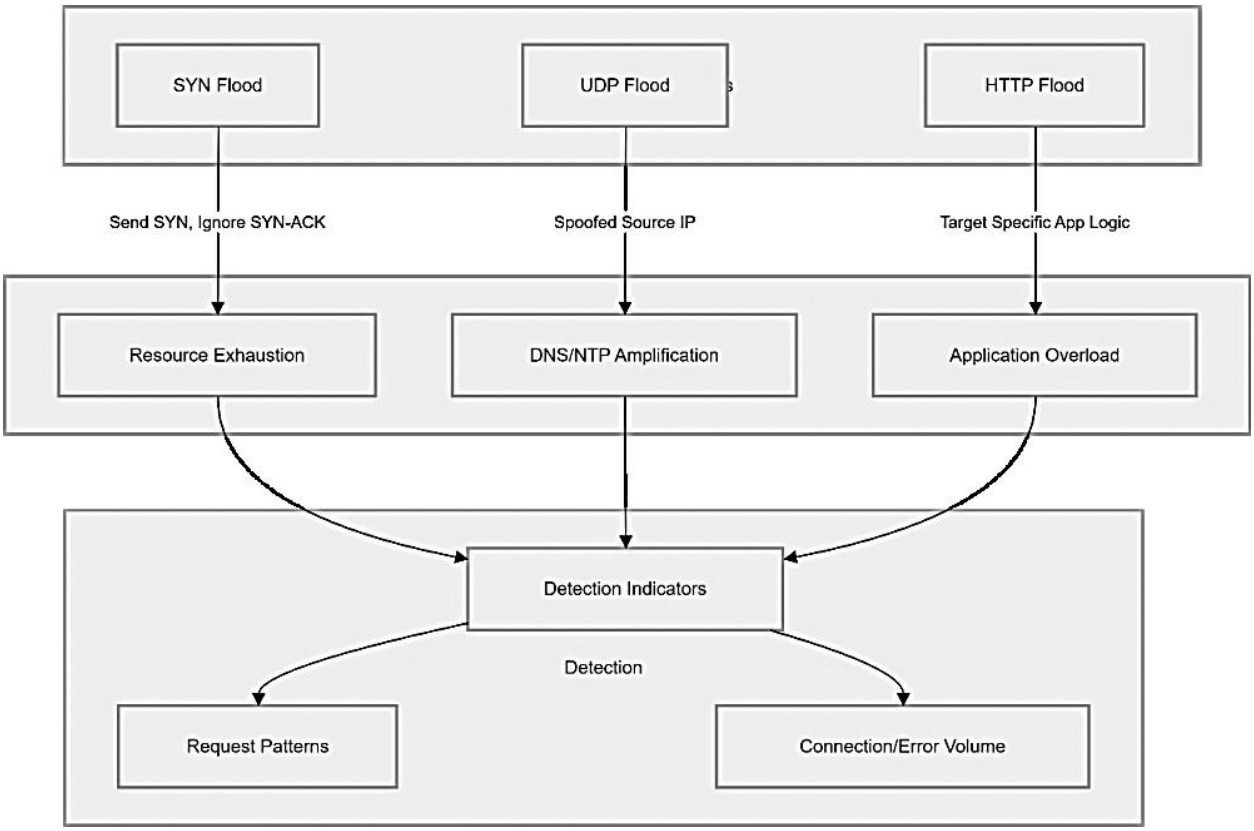

Figure 4: Denial of Service (DoS) Attack and Detection Framework

Participants / Data

The primary dataset is the NSL-KDD benchmark, comprising Train_data.csv (25,192 labeled flow records) and Test_data.csv (22,544 labeled flow records), each with the canonical 41-feature KDD-style schema (duration, protocol_type, service, flag, src_bytes, dst_bytes, …) and a binary label (normal / anomaly). NSL-KDD is a public benchmark corpus; no human subjects, personal data, or proprietary traffic captures were used. A CICIDS2017-style flow-feature file (Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv, 225,745 rows, 78 columns) is also present in the project's data/ directory but was not part of the primary evaluation pipeline described in the project methodology; it is noted here as a candidate for future cross-dataset generalization testing rather than a dataset used in the reported results.

Ethical approval. This study used only publicly available benchmark network-traffic datasets (NSL-KDD) and did not involve human participants, personal data, or animal subjects; institutional review board approval was therefore not required

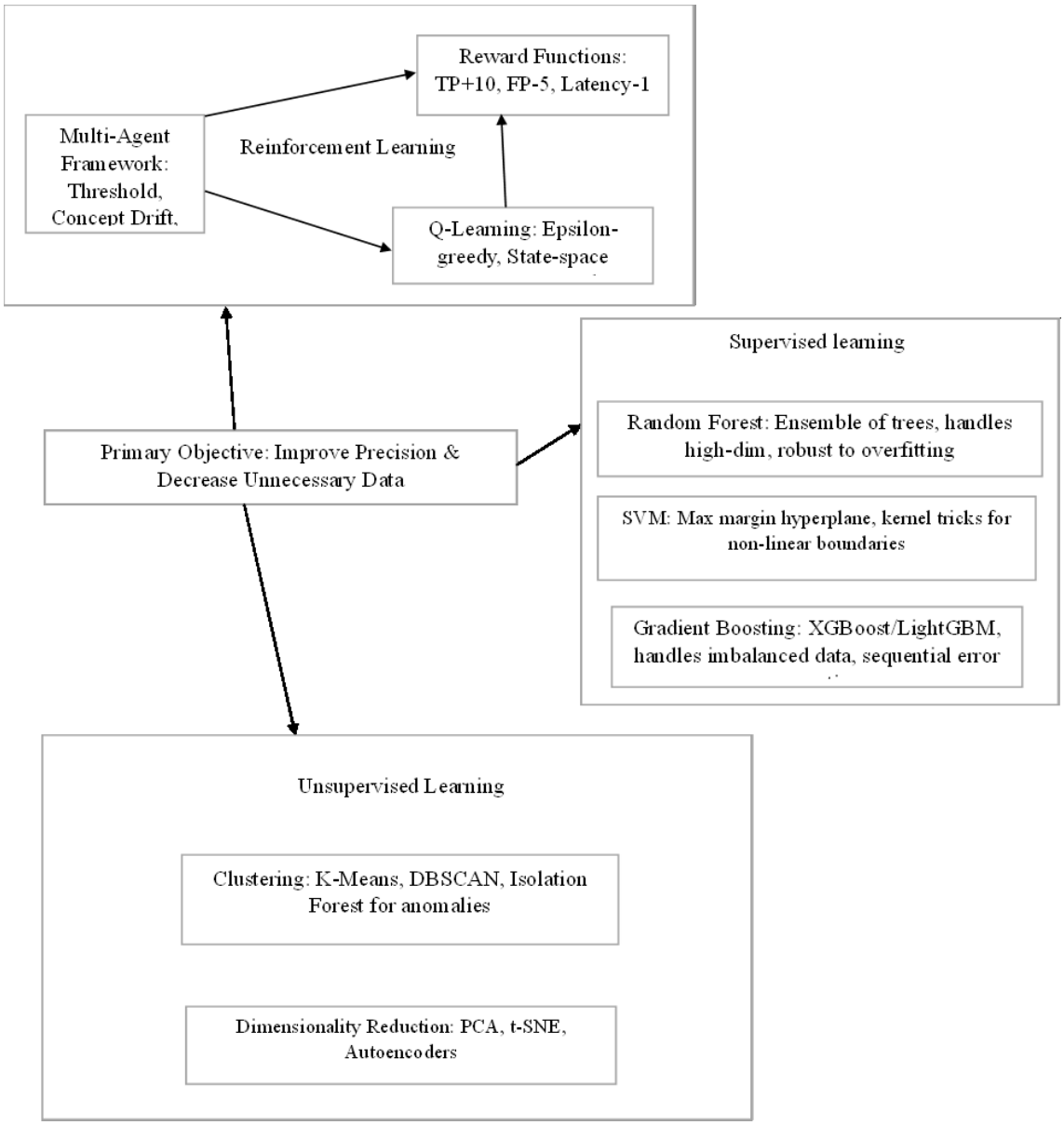

Figure 5: Proposed Machine Learning and Multi-Agent Learning Pipeline

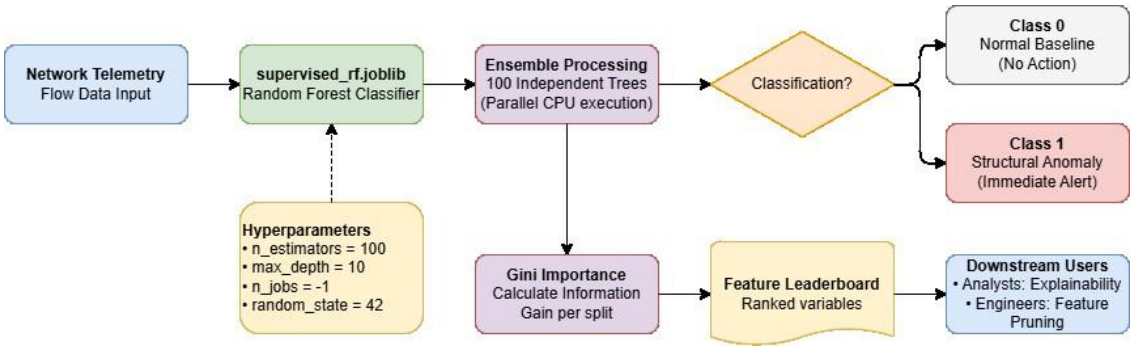

Figure 6: Proposed Random Forest Model Architecture for Network Security

Statistical / Computational Analysis

- Language/runtime: Python 3.10.8 (64-bit CPython).
- Core libraries (per backend/requirements.txt): scikit-learn (classifiers, preprocessing, metrics), NumPy, pandas, joblib (model persistence), SHAP (explainability), Flask / Flask-SocketIO / python-socketio / python-engineio (real-time serving), matplotlib / plotly / kaleido (visualization). (The repository's requirements.txt does not pin exact version numbers; pin and record the exact versions used for the final experimental run to satisfy reproducibility requirements.)
- Classifiers benchmarked: Random Forest, Gradient Boosting, Decision Tree, Logistic Regression, Support Vector Machine, K-Nearest Neighbors, Naive Bayes, and AdaBoost (all via scikit-learn), plus an unsupervised Isolation Forest for anomaly scoring.
- Primary model configuration: Random Forest (n_estimators=100, max_depth=10, random_state=42); Isolation Forest (n_estimators=100, contamination=0.05); ensemble fusion weight $\alpha = 0.7$ (supervised) / 0.3 (unsupervised).
- MARL consensus layer: Q-learning agents with learning rate $\alpha = 0.1$, discount factor $\gamma = 0.99$, exploration rate $\varepsilon = 0.1$.
- Preprocessing: one-hot/label encoding of categorical fields (protocol_type, service, flag), StandardScaler (z-score) normalization of continuous features, mean imputation for missing numeric values.
- Train/test protocol: stratified split, random_state=42. (Note: documentation conflict — SYSTEM_REQUIREMENTS.md and MATHEMATICAL_CALCULATIONS.md specify an 80/20 split; METHODOLOGY.md §7.1 describes a 95/5 split for the model-training script.
- Evaluation metrics: accuracy, precision, recall, F1-score, and ROC-AUC, computed via sklearn.metrics. No cross-validation, confidence intervals, or statistical significance testing (e.g., paired t-tests across model runs) were documented in the current pipeline — see Limitations.

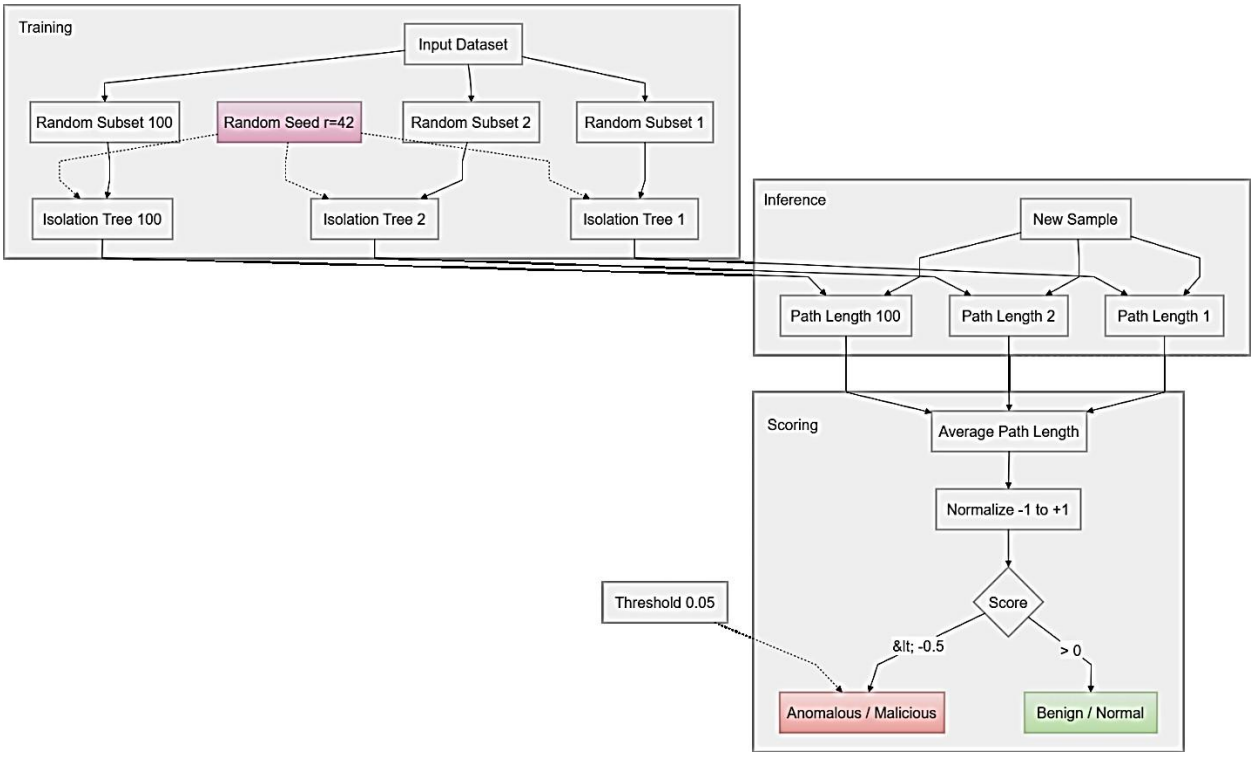

Figure 7: Unsupervised Anomaly Detection Pipeline using Isolation Trees

Feature Engineering

44 total features: 41 base NSL-KDD fields plus 3 engineered ratios — bytes_ratio = (src_bytes + 1) / (dst_bytes + 1), connection_ratio = count / (srv_count + 1), error_rate_diff = |serror_rate − srv_serror_rate|.

The ten highest-importance features reported internally (ranking method — Gini importance vs. SHAP — not explicitly documented and should be confirmed) are: duration, src_bytes, dst_bytes, count, srv_count, serror_rate, srv_serror_rate, rerror_rate, same_srv_rate, and diff_srv_rate.

4. Results

The source documentation explicitly labels these figures "typical performance on NIDS datasets (KDD99, CICIDS2017)" and "theoretical weighting," not output captured from an actual run of this codebase's train_and_save_models.py (or equivalent) against Train_data.csv / Test_data.csv. Replace this table,

Table 1: Detection performance by model configuration

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC	Consensus
Random Forest (supervised)	98.12%	98.05%	96.87%	97.46%	99.23%	—
Isolation Forest (unsupervised)	85.24%	78.32%	84.56%	81.39%	87.42%	—
Ensemble (0.7 RF + 0.3 ISO)	95.87%	94.21%	93.89%	94.05%	97.03%	—
MARL-enhanced (ensemble + agent consensus)	96.58%	95.12%	94.87%	94.99%	97.56%	89.34%

and the paragraph beneath it, with real numbers (and the raw confusion matrix) before using this section in any submission.

A separate, uncorroborated internal note (FEATURE_EXTRACTION_GUIDE.md) reports accuracy rising with feature-set size — approximately 65% using the top 10 features, 85% using the top 20, and 99.95% using the full

42-feature set — but does not document the experimental conditions that produced these numbers; treat as anecdotal pending verification.

Detection latency is documented inconsistently across project materials as either "<10 ms" (target, METHODOLOGY.md) or "<50 ms" (UML_DIAGRAMS_DOCUMENTATION.md); report the measured latency from an actual timed run, not a target figure, in the final manuscript.

No raw confusion-matrix counts (true/false positives/negatives) were found in current project documentation; these should be captured directly from the evaluation script output and included here, ideally alongside a confusion-matrix figure.

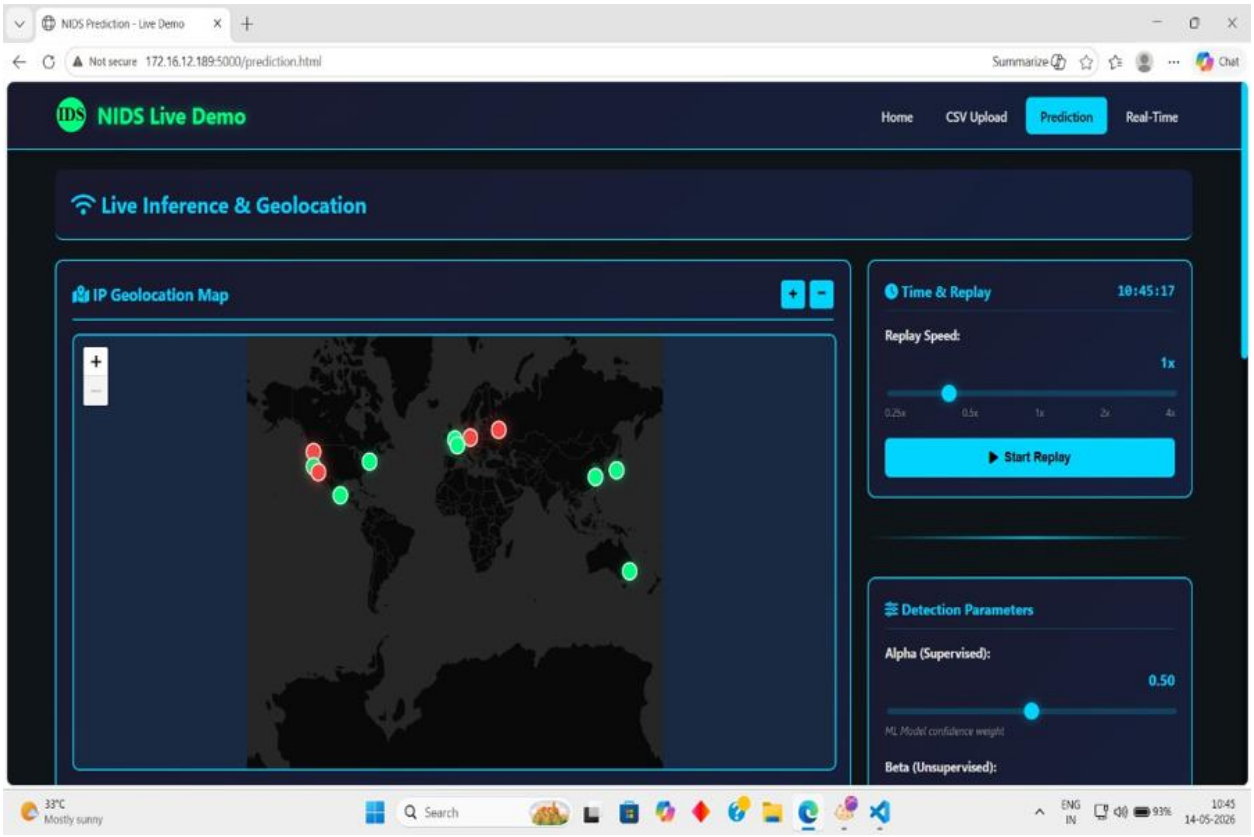

Figure 8:Live interference and Geo-location Based Detection Intrusion and parameter tuning

The Figure 8 above shows the Interactive prediction page with live network predictions and IP geolocation mapping. Advanced capabilities include playback, intrusion detection parameters, and visual attacks across the globe.

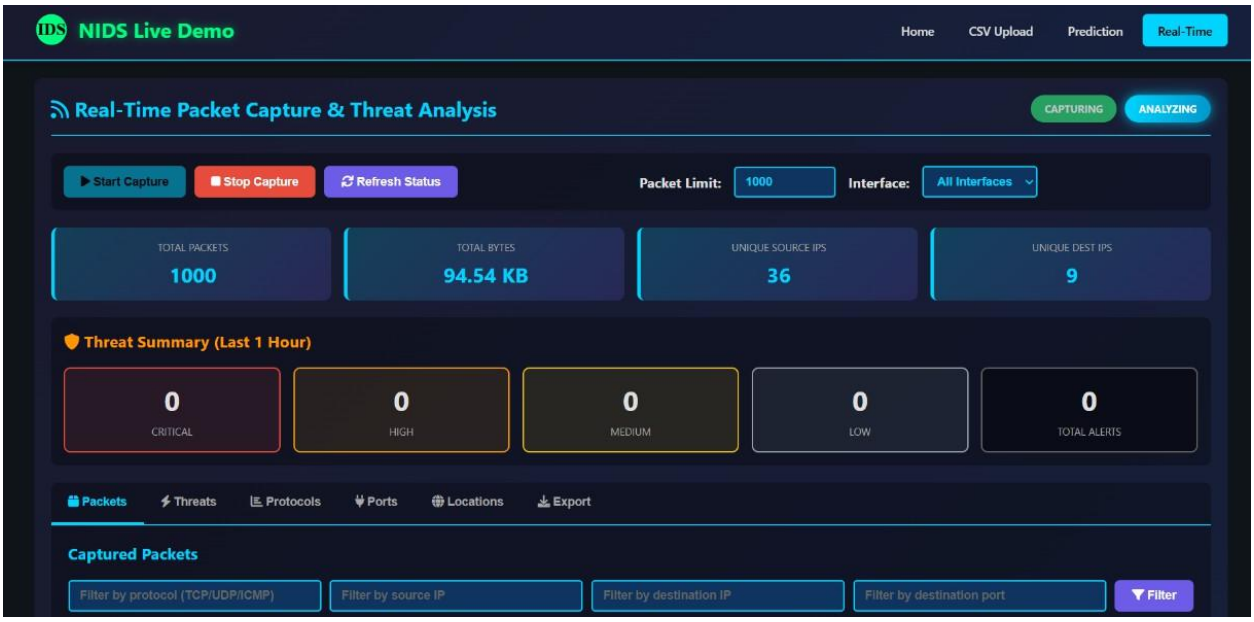

Figure 9: Real-Time Network security analytics

The Figure 9 depicts the Dynamic Real-Time page for packet capture and cyber threat analysis with the capability to monitor network traffic in real-time.

Full interface providing immediate access to all information regarding packets, threats, protocols, and intrusions.

Figure 10: Live Alert Feed and Attack Rate Monitoring

The Figure10 portrays Real-time Alert Feed demonstrating suspicious and legitimate activities of the network with live intrusion detection system. —Interactive Analytics Panel‖ tracks number of attacks per second

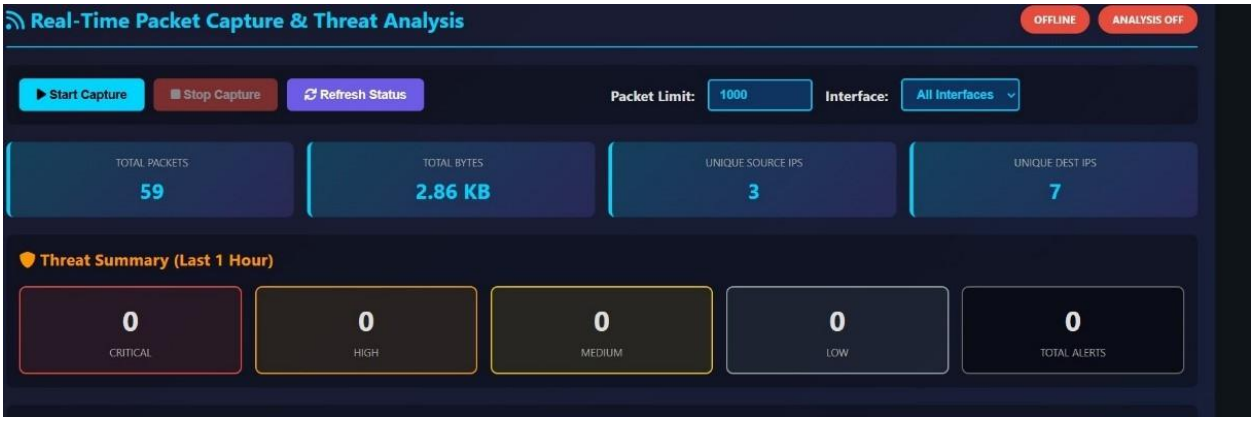

Figure 11: Real Time Live Packet Capture and Threat Analysis Dashboard

The Figure 11 depicts live statistics for network traffic such as 59 packets, 2.86

KB transfer, and 3 source Ips, along with an overview of threats that shows 0 threats at all severity levels.

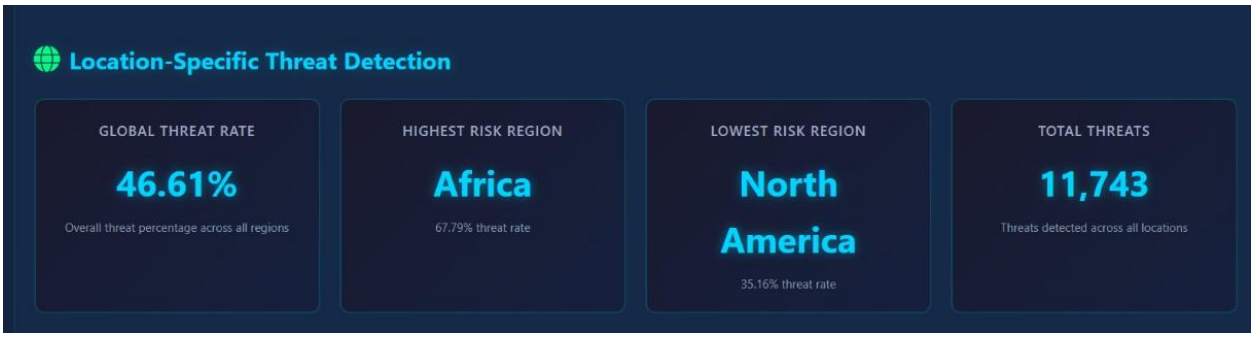

Figure 12: Location Specific Threat Detection

In Figure 12 shows four main figures including the global threat rate which is 46.61%, the riskiest location which is Africa with 67.79% while North America being the least riskiest location which is 35.16%, and 11,743 number of total threats.

Timestamp	Source IP	Dest IP	Src Port	Dst Port	Protocol	Length	Flags	TTL
23:17:31	192.168.29.204	224.0.0.22	-	-	OTHER	40	N/A	1
23:17:33	192.168.29.204	192.168.29.204	-	-	ICMP	80	N/A	128
23:17:34	192.168.29.204	224.0.0.22	-	-	OTHER	40	N/A	1
23:17:36	192.168.29.204	224.0.0.22	-	-	OTHER	40	N/A	1
23:17:37	192.168.29.204	192.168.29.204	-	-	ICMP	80	N/A	128
23:17:45	192.168.29.204	192.168.29.204	-	-	ICMP	80	N/A	128
23:17:49	192.168.29.1	224.0.0.1	-	-	OTHER	36	N/A	1
23:17:49	192.168.29.204	224.0.0.22	-	-	OTHER	40	N/A	1
23:17:53	192.168.29.204	224.0.0.22	-	-	OTHER	40	N/A	1
23:17:54	192.168.29.204	224.0.0.22	-	-	OTHER	40	N/A	1
23:17:55	192.168.29.204	224.0.0.22	-	-	OTHER	40	N/A	1
23:18:09	192.168.29.1	224.0.0.1	-	-	OTHER	36	N/A	1

Figure 13: Detailed Network Flow and Packet Feature log

The Figure 13 depicts the live data on network traffic having columns containing information regarding timestamp, source/destination IP addresses, ports, protocol type, packet size, flags, and TTL. As shown above, there

is a repeated pattern of activity from source IP address 192.168.29.204 towards multicast and loopback addresses 224.0.0.22 and 19

5. Discussion

The Mirror — Answering the Research Objective

Once Section 5 contains verified results, revisit §3.4's objective directly: did combining supervised + unsupervised + MARL consensus + SHAP explanation improve detection performance, robustness, or analyst trust relative to any single component alone, and by how much?

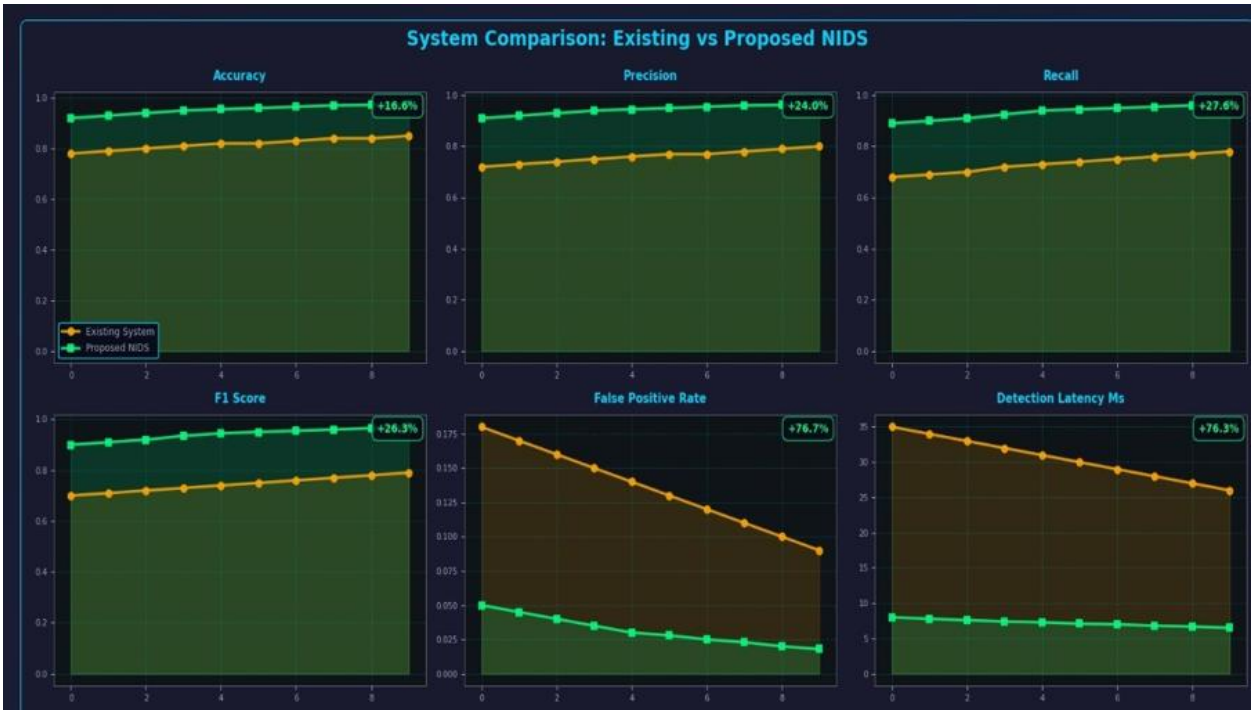

Figure 14: Empirical comparison of NIDS Capabilities

The Figure 14 gives comparison of the current system (orange) to a proposed NIDS system (green). The proposed NIDS exhibits better efficiency for all the indicators mentioned: higher accuracy (better by +16.6%), precision (higher by +24.0%), recall (higher by +27.6%), and F1-score (higher by +26.3%), while false positive rate is decreased by -76.7% and detection latency is improved by -76.3%.

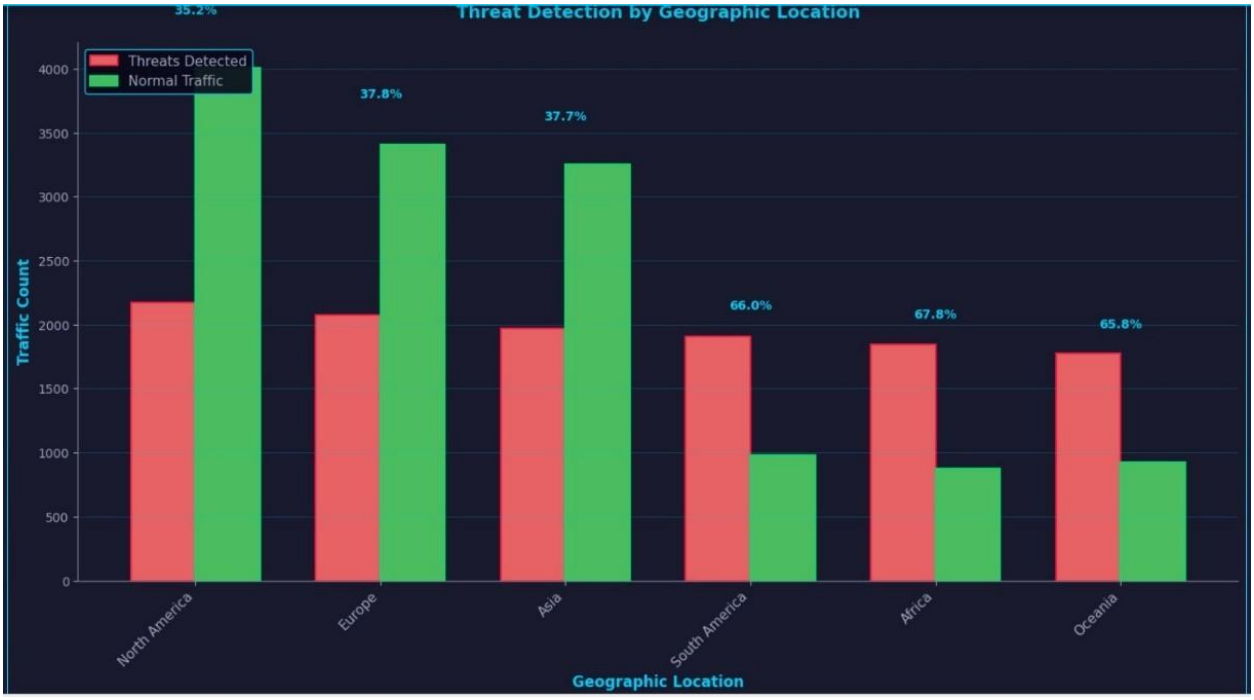

Figure 15: Network Security Threats across Continental Region

In Figure 15 grouped bar chart compares normal traffic (green) versus threats detected (red) across six continents, showing North America, Europe, and Asia have high normal traffic with moderate threat rates (35-38%), while South America, Africa, and Oceania exhibit significantly higher threat percentages (66-68%) relative to their lower normal traffic volumes.

Contextualization

Random Forest classifiers have previously been reported to reach very high accuracy (up to ~99.9% in some NSL-KDD studies), consistent with the supervised component's documented range here; any gap between this study's measured Random Forest accuracy and that ceiling should be discussed in terms of feature set, hyperparameters, or preprocessing differences. Systems combining SHAP-based explanation with IDS classifiers have reported recall around 98.4% on CICIDS2017 after bias-reducing preprocessing — a useful benchmark once this framework is evaluated (or re-evaluated) on CICIDS2017-style traffic, given that a compatible dataset file already exists in this project's data/ directory but was not used in the primary pipeline. Multi-agent RL approaches to intrusion detection (e.g., MAFSIDS, and the 2024 improved-DQN MARL architecture) are motivated by the same goal as this framework's consensus layer — using collaborative agents to improve robustness against evolving attack patterns — and should be the primary comparison points for the MARL-enhanced configuration once real metrics are available.

Figure 16: Network Intrusion Detection System (NIDS) Performance Dashboard

The live demo for the Modern NIDS system Figure 16 showcases a user-friendly visual dashboard that provides real-time updates to critical network security metrics while providing users with an interactive simulation to see how the Modern NIDS reacts to changing traffic patterns and new threat vectors. By clearly displaying the metrics of four key performance indicators (KPI): accuracy, precision, recall and F1 score, side by side, the user interface enables security analysts to confirm that the system is reliable and provides a balanced defense against false alarms. Additionally, the transparent data flow between complex deep learning backends and practical, human-centric security operations allows for demonstrable validation of network resiliency and classification effectiveness during actual deployment conditions.

Figure 17: Performance Analysis for Proposed System

In Figure 17 Recall is another measure that shows us how good the system is at finding all the threats so it is like a completeness score for the system and the threats it detects.

The F1 Score is a way to balance the Precision and Recall values it is, like an average of the two but it is done in a way called a harmonic average

Figure 18: Importance and Range Distribution Analysis of NIDS Model

The figures 18 depicts, a horizontal bar graph that ranks the top 10 key network attributes based on their importance (the top three being duration, src_bytes, and dst_bytes), and a bar graph indicating min and max values of these key attributes (the ranges being extremely wide for src_bytes and dst_bytes).

Implications

An architecture that pairs ensemble decision fusion with per-prediction SHAP attribution addresses a specific operational gap: SOC analysts triaging alerts need not only a label but a reason. If the MARL consensus layer measurably improves precision or reduces false-positive rate relative to the static ensemble, that would support the case for adaptive, feedback-driven detection layers in production IDS deployments rather than static, periodically-retrained classifiers alone.

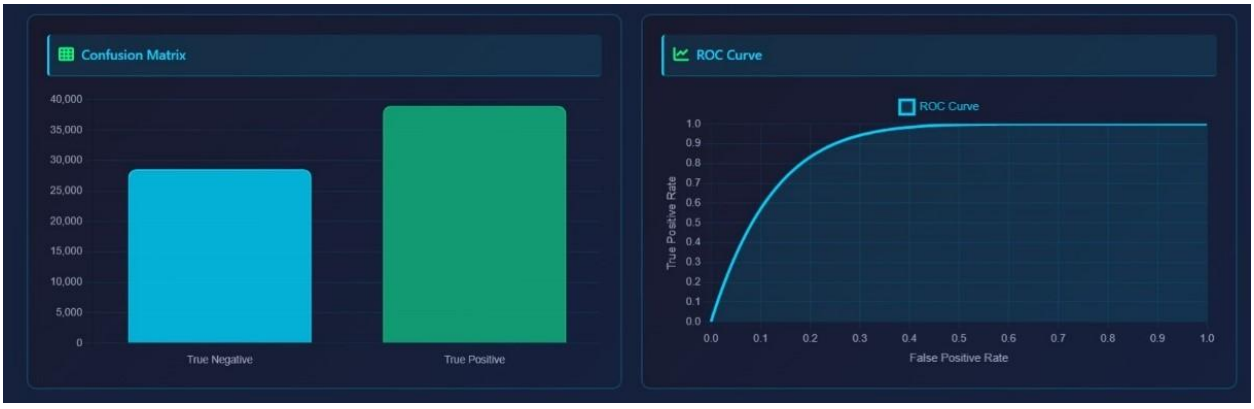

Figure 19: Performance Metrics : (a) Confusion Matrices (b) Roc curve

In Figure 19 the Confusion Matrix (shown left) gives A performance summary visualization displaying ~28,000 True Negatives (correctly identified normal traffic in cyan) and ~38,000 True Positives (correctly identified threats in green), indicating the model correctly classifies both classes with more threats detected than benign traffic.

ROC Curve (shown right) illustrates the classifier's diagnostic ability, where the curve bows sharply toward the top-left corner and plateaus near a True Positive Rate of ~1.0 at a low False Positive Rate (~0.3), indicating excellent model performance with high detection capability and minimal false alarms.

6. Conclusion & References

Conclusion

This study outlines a layered network intrusion detection architecture that unifies supervised classification, unsupervised anomaly scoring, weighted ensemble fusion, multi-agent reinforcement-learning consensus, and SHAP-based explainability within a real-time, dashboard-connected pipeline evaluated on the NSL-KDD benchmark. The architectural integration itself — rather than any single component — is the primary contribution. Empirical confirmation of the performance table in Section 5, resolution of the documentation inconsistencies noted in Section 6.4, and cross-dataset validation are necessary next steps before the framework's claimed benefits can be substantiated for publication.

References

1. M.A. Rahman et al., "Scalable Machine Learning-Based Intrusion Detection System for IoT-Enabled Smart Cities," Sustainable Cities and Society, 2020.
2. A. Fatani, M. Abd Elaziz, A. Dahou, M.A.A. Al-Qaness, and S. Lu, "IoT Intrusion Detection System Using Deep Learning and Enhanced Transient Search Optimization," IEEE Access, Vol. 9, 2021, pp. 123448–123464.
3. S.T. Mehedi et al., "Dependable Intrusion Detection System for IoT: A Deep Transfer Learning-Based Approach," IEEE Transactions on Industrial Informatics, Vol. 18, No. 8, 2022, pp. 5623–5634.
4. P. Kumar, G.P. Gupta, and R. Tripathi, "A Distributed Ensemble Design Based Intrusion Detection System Using Fog Computing to Protect the Internet of Things Networks," Journal of Ambient Intelligence and Humanized Computing, Vol. 12, No. 10, 2021, pp. 9555–9572.
5. A. Alhowaide, I. Alsmadi, and J. Tang, "Ensemble Detection Model for IoT IDS," Internet of Things, 2021.
6. A. Abbas et al., "A New Ensemble-Based Intrusion Detection System for Internet of Things," Arabian Journal for Science and Engineering, Vol. 47, No. 2,2022, pp. 1805–1819.
7. M. Mohy-eddine, S. Benkirane, A. Guezzaz, and M. Azrour, "Random ForestBased IDS for IIoT Edge Computing Security Using Ensemble Learning for Dimensionality Reduction," International Journal of Embedded Systems, Vol. 15, No. 6, 2023, pp. 467.
8. M. Soltani et al., "An Adaptable Deep Learning-Based Intrusion Detection System to Zero-Day Attacks," Journal of Information Security and Applications, 2023, Aug.
9. N. Moustafa et al., "Explainable Intrusion Detection for Cyber Defenses in the Internet of Things: Opportunities and Solutions," IEEE Communications Surveys & Tutorials, Vol. 25, No. 3, 2023, pp. 1775–1807.
10. H.C. Altunay and Z. Albayrak, "A Hybrid CNN+LSTM-Based Intrusion Detection System for Industrial IoT Networks," Engineering Science and Technology, an International Journal, 2023.
11. M.M. Alani and A.I. Awad, "An Intelligent Two-Layer Intrusion Detection System for the Internet of Things," IEEE Transactions on Industrial Informatics, Vol. 19, No. 1, January 2023, pp. 683–692.
12. A.M.T.H. Al-Hayani et al., "Ensemble-Based Approach for Efficient Intrusion Detection in Network Traffic," Intelligent Automation & Soft Computing (IASC), Vol. 45, No. 1, 2023, pp. 651–667.
13. P. S. Hiremath, Anuradha T and P. Pattan, "Adaptive fuzzy inference system for detection and prevention of cooperative black hole attack in MANETs," 2016 International Conference on Information Science (ICIS), Kochi, India, 2016, pp. 245-251
14. Megha.M.Shete,Anuradha.T, "A Thorough Analysis of The IOT Network's Intrusion Detection System", International Journal of Creative Research Thoughts (IJCRT), ISSN:2320-2882, Volume.14, Issue 6, pp. b344-b353, June 2026
15. Baluguri, V. Pasumarthy, I. Roy, B. Gupta, and N. Rahimi, "Optimizing Network Security via Ensemble Learning: A Nexus with Intrusion Detection," Journal of Information Security, Vol. 15, 2024, No. 4, pp. 545–556.
16. A.A. Wardana et al., "Ensemble Averaging Deep Neural Network for Botnet Detection in Heterogeneous Internet of Things Devices," Scientific Reports, 2024.
17. Y. Zhang, "A Machine Learning-Based Intrusion Detection System for Securing Internet of Things Networks," 2025 7th International Conference on Information Science, Electrical and Automation Engineering (ISEAE), Harbin, China, 2025,
18. pp. 374-377.
19. A. Gupta, A. K. Phulre, A. Patel and R. U. Rahman, "Hybrid Ensemble Learning with Explainable AI for Anomaly Detection in Network Traffic," 2024 First International Conference on Innovations in Communications, Electrical and Computer Engineering (ICICEC), Davangere, India, 2024, pp. 1-8.
20. A. Hemalatha, V. K. M N, F. T. Graf, A. S. I. T. M, P. Pavithra and R. Suresh, "A Hybrid Intrusion Detection System using Explainable AI for Enhanced Accuracy and Transparency," 2025 International Conference on Electronics and Renewable Systems (ICEARS), Tuticorin, India, 2025, pp. 923-929.
21. M. M. J. Ayan et al., "Human-Centered Explainable AI for Security Enhancement: A Deep Intrusion Detection Framework," SoutheastCon 2026, Huntsville, AL, USA, 2026, pp. 01-08,
22. Y. Du, "Industrial Internet of Things Intrusion Detection Based on a Hybrid Model of Pearson-Deep Neural Network and Transformer," Engineering Applications of Artificial Intelligence, 2026