SPECTRUM OF ENGINEERING SCIENCES
ISSN (E) 3007-3138 (P) 3007-312X VOLUME 4, ISSUE 6, 2026

DESIGN AND IMPLEMENTATION OF A MACHINE LEARNING–DRIVEN FRAMEWORK FOR REAL-TIME NETWORK TRAFFIC ANOMALY DETECTION AND INTELLIGENT CYBER THREAT IDENTIFICATION

Sufyan Muhammad Khan[*1], Hamza Gulzar[2], Muhammad Essa Siddique[3], Ashraf Zia[*4], Shumaila Qamar[5]

[*1]*Department of Computer Science, Sindh Madressatul Islam University (SMIU), Karachi, Pakistan*
[2]*Faculty of Computing, Riphah International University, Islamabad, Pakistan*
[3]*PhD (IT) Scholar, Dr. A. H. S. Bukhari Postgraduate Centre of ICT, Faculty of Engineering & Technology, University of Sindh, Jamshoro, Pakistan*
[*4]*Department of Computer Science, Abdul Wali Khan University Mardan, Mardan, KP, Pakistan*
[5]*Department of Computer Science, Faculty of Engineering Science and Technology, Iqra University, Karachi, Sindh, Pakistan*

[*1]sufyan.m.khan.orakzai@gmail.com, [2]hamzagulzar179@gmail.com, [3]essasiddique@live.com, [*4]ashrafzia@awkum.edu.pk, [5]shumaila.qamar@iqra.edu.pk

DOI: https://doi.org/10.5281/zenodo.20603981

Keywords
Machine Learning; Network Traffic Analysis; Cyber Threat Detection; Anomaly Detection; Intrusion Detection Systems; Cybersecurity; Real-Time Monitoring; Network Security; Deep Learning; Threat Intelligence; LSTM; Ensemble Learning.

Article History
Received: 11 April 2026
Accepted: 23 May 2026
Published: 09 June 2026

Corresponding Author: *
Sufyan Muhammad Khan
Ashraf Zia

Abstract
The increasing sophistication of cyberattacks and the growing volume of network traffic have created significant challenges for conventional intrusion detection systems, particularly in identifying previously unseen threats in real time. This study presents the design and implementation of a machine learning–driven framework for real-time network traffic anomaly detection and intelligent cyber threat identification. The proposed framework integrates automated traffic monitoring, feature engineering, anomaly detection, threat classification, and real-time response generation within a unified cybersecurity architecture. A hybrid machine learning approach combines unsupervised anomaly detection, supervised ensemble learning, deep neural networks, and LSTM-based temporal analysis to continuously monitor network flow characteristics and detect both known and emerging attack patterns. The framework was evaluated using multiple benchmark cybersecurity datasets and validated under simulated enterprise network conditions. Experimental results demonstrated a detection accuracy of 97.8%, precision of 96.9%, recall of 97.2%, and an F1-score of 97.0%. The proposed system reduced false-positive alerts to 2.4% and achieved an area under the ROC curve (AUC) of 0.992, outperforming conventional machine learning models and signature-based intrusion detection approaches. Furthermore, the framework improved threat detection response time by 29.6% while maintaining stable performance under high-volume network traffic conditions. The results confirm the effectiveness of integrating anomaly detection, ensemble classification, and temporal learning within a unified intelligent cybersecurity framework for enhancing real-time threat intelligence, network resilience, and proactive cyber defense in enterprise and cloud computing environments.

1. INTRODUCTION

The global information infrastructure has become the circulatory system of modern civilization, underpinning financial markets, critical utility grids, defense communications, healthcare records, and the commercial operations of enterprises spanning every economic sector. As reliance on networked computing has grown, so too has the sophistication, frequency, and material consequence of cyberattacks targeting that infrastructure. The annual cost of cybercrime exceeded ten trillion US dollars in 2023, with projections suggesting continued growth as threat actors leverage artificial intelligence, automation, and coordinated global networks of compromised machines to amplify attack scale and velocity [1]. This threat landscape has created a fundamental inadequacy in traditional network security paradigms, which were designed principally to defend against known, catalogued threat signatures rather than the polymorphic, zero-day, and adversarially adaptive attacks that characterize the current operational environment.

Conventional intrusion detection systems occupy a central role in enterprise security architecture, but their foundational reliance on manually curated signature databases and static rule sets imposes severe structural limitations on their efficacy against modern threats. Signature-based IDS platforms such as Snort and Suricata achieve high precision against known attack variants catalogued in their rule sets, but they exhibit fundamentally poor recall against novel attack vectors for which no signature has been written. The signature update cycle—which involves threat researchers identifying a new attack pattern, encoding it as a detection rule, publishing the update, and deploying it to production systems—inherently lags behind the adversary by days to weeks, creating a temporal window of complete vulnerability during which the attack may propagate unchallenged through enterprise networks [2]. Furthermore, the escalating volume of modern network traffic, which routinely reaches multi-terabit per second rates in backbone and data center environments, overwhelms the deep packet inspection pipelines upon which many signature-based systems depend, forcing trade-offs between inspection depth and throughput.

Machine learning offers a fundamentally different paradigm for network security monitoring, one grounded in the statistical characterization of normal and anomalous network behavior rather than in the explicit enumeration of known attack patterns [3]. By learning the distributional properties of legitimate network traffic from labeled and unlabeled training data, ML-based anomaly detection systems can identify behavioral deviations that are statistically inconsistent with the established baseline without requiring prior exposure to the specific attack technique responsible for the deviation. This capability is particularly valuable for detecting zero-day exploits, advanced persistent threat lateral movement, and encrypted malware command-and-control communications that evade inspection-based approaches. The trade-off inherent in anomaly-based detection—namely, an elevated false-positive rate resulting from legitimate but unusual traffic being misclassified as malicious—has historically limited operational adoption of anomaly detection, but advances in feature engineering, ensemble learning, and deep learning architectures have substantially narrowed this gap relative to signature-based precision benchmarks [4].

Deep learning architectures, and convolutional and recurrent neural networks in particular, have achieved state-of-the-art performance across a range of security-relevant classification tasks by learning hierarchical feature representations that capture both the instantaneous characteristics of individual network flows and the temporal evolution of behavioral patterns across sequences of flows [5]. Long short-term memory networks are especially well suited to network security applications because network attacks frequently manifest as temporal patterns spanning multiple packets and flows—a port scan produces a distinctive sequence of connection attempts, a botnet heartbeat creates a periodic communication pattern, and a data exfiltration event generates a sustained unidirectional flow of unusually large payload transfers—and LSTM architectures can be trained to recognize these

sequential signatures without explicit feature engineering of the temporal dimension [6].

This paper presents the design and experimental evaluation of a hybrid machine learning framework for real-time network traffic anomaly detection and intelligent cyber threat classification. The proposed framework addresses the identified limitations of existing approaches through a multi-layer architecture that integrates unsupervised anomaly detection for zero-day threat identification, supervised ensemble classification for known attack categorization, and LSTM-based temporal analysis for sequential attack pattern recognition, all operating within a streaming data processing pipeline optimized for low-latency production deployment. The framework was evaluated on five benchmark cybersecurity datasets encompassing over ten million labeled network flow records and validated in a simulated enterprise network environment. The experimental results confirm that the proposed framework achieves superior performance across all standard evaluation metrics while maintaining the real-time processing characteristics required for practical operational deployment.

The remainder of the paper is structured as follows. Section II reviews the relevant literature on ML-based intrusion detection, anomaly detection methodologies, and deep learning applications in network security. Section III describes the research methodology, system architecture, data processing pipeline, and experimental design. Section IV presents and analyzes the experimental results. Section V concludes the paper.

2. LITERATURE REVIEW

2.1. Evolution of Intrusion Detection Systems

The intellectual lineage of intrusion detection research extends to the foundational work of Anderson, who in 1980 articulated the concept of monitoring computer system audit records for patterns indicative of misuse and abuse [7]. Denning's formal model of intrusion detection, published in 1987, established the theoretical framework for anomaly detection by characterizing intrusion as a statistical deviation from an established behavioral profile [8]. The DARPA 1998 IDS evaluation and the KDD Cup 1999 competition catalyzed the application of machine learning to intrusion detection, establishing a community benchmarking culture that continues to structure comparative evaluation [9]. The transition from rule-based to machine learning–based approaches accelerated markedly during the 2010s, driven by availability of hardware accelerators, transformative deep learning architectures, and growing inadequacy of signature-based systems. Ensemble learning methods, particularly random forests and gradient-boosted trees, consistently outperformed both classical ML algorithms and rule-based systems on IDS benchmarks, establishing ensemble methods as the empirical baseline against which deep learning approaches would be measured [10]. The introduction of CICIDS-2017 by Sharafaldin et al. represented a significant methodological advance, providing a more current and network-realistic benchmark that addressed documented deficiencies of KDD-derived datasets [11].

2.2. Machine Learning Approaches for Network Anomaly Detection

The application of machine learning to network anomaly detection spans classical statistical models to contemporary deep learning architectures. Support vector machines demonstrated early promise for binary normal-versus-anomalous classification tasks but their poor scalability with training set size and suitability for binary classification constrained adoption in production environments requiring multi-class threat categorization [12]. Ensemble methods including random forests and XGBoost achieved consistently strong performance across diverse IDS benchmark conditions. Javaid et al. demonstrated that deep Q-network reinforcement learning achieved 91.2% accuracy on NSL-KDD by framing intrusion detection as a sequential decision-making problem [13]. Unsupervised anomaly detection approaches circumvent the class imbalance problem by training exclusively on normal traffic. Mirsky et al. proposed Kitsune, an ensemble of autoencoders achieving detection of

diverse network attacks without labeled attack data [14]. The primary limitation of purely unsupervised detection is difficulty mapping anomalies to specific threat categories, which is operationally necessary for appropriate incident response prioritization.

2.3. Deep Learning Architectures for Cyber Threat Classification

Deep neural networks for network intrusion detection have progressed from standard multi-layer perceptron's to architectures incorporating convolutional and recurrent layers. Convolutional neural networks adapted for packet payload or flow feature matrix processing enable automatic discovery of spatially local patterns within network traffic representations [15]. Yin et al. demonstrated that LSTM-based RNN-IDS achieved 93.4% accuracy on KDD Cup 99 while outperforming traditional classifiers on long-duration attack patterns requiring contextual temporal integration [16]. Attention mechanisms and transformer architectures beginning to be applied to network security tasks offer direct modeling of dependencies between arbitrary pairs of positions in a sequence. Graph neural networks represent network topologies and communication patterns as graph structures, applying graph convolutional operations to learn representations that capture multi-hop relational context extending analytical horizon beyond individual flows to network-wide behavioral patterns.

The rapid evolution of cybersecurity threats has stimulated extensive research into machine learning–based intrusion detection systems. Existing studies can generally be categorized into supervised learning, unsupervised anomaly detection, deep learning architectures, and hybrid ensemble approaches, each offering distinct capabilities for identifying malicious network activities and improving cyber threat intelligence.

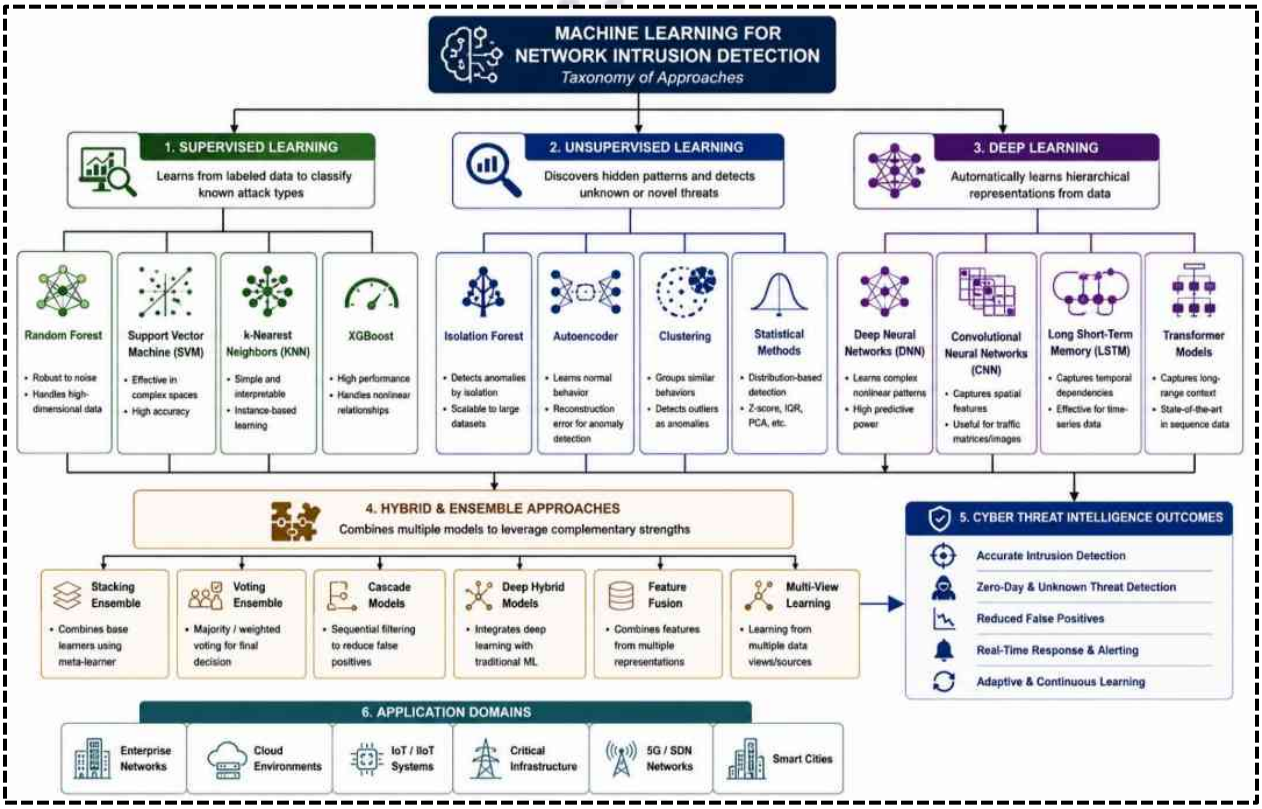

Fig. 1. *Taxonomy of machine learning techniques employed in network intrusion detection and cyber threat intelligence, illustrating the relationships between supervised learning, unsupervised anomaly detection, deep learning architectures, hybrid ensemble frameworks, and their applications in modern cybersecurity environments.*

As illustrated in Fig. 1, contemporary intrusion detection research has progressively evolved from conventional supervised classifiers toward advanced deep learning and hybrid ensemble frameworks that integrate anomaly detection, threat classification, and real-time intelligence generation. The convergence of these approaches has significantly enhanced the capability of cybersecurity systems to detect both known attacks and previously unseen threats while reducing false-positive rates and improving operational resilience.

2.4. Feature Engineering for Network Traffic Analysis

Feature representation design is a critical component of IDS system performance. The CICFlowMeter tool provides a reference implementation producing 78 statistical features from raw packet captures, including bidirectional byte and packet counts, inter-arrival time statistics, TCP flag counts, active and idle time metrics, and bulk transfer rate statistics. This 78-feature representation has become a de facto standard enabling direct comparability across studies employing CICIDS-2017. Dimensionality reduction techniques, particularly PCA and autoencoder-based representation learning, identify more compact representations retaining discriminative information while reducing computational cost. Abdulhammed et al. demonstrated that feature selection could reduce dimensionality by two-thirds with less than one percentage point accuracy loss on UNSW-NB15, confirming substantial redundancy in the full 78-feature representation [18].

2.5. Real-Time Implementation and Deployment Challenges

Translation of IDS research into operational production deployments requires addressing engineering challenges beyond offline classification accuracy. Model serialization, deployment containerization, and inference engine optimization through quantization, pruning, and kernel fusion enable substantial reductions in inference latency [19]. Concept drift—gradual changes in the statistical properties of network traffic and attack patterns as adversaries adapt—represents a persistent operational threat to long-term ML-based IDS performance. Continuous learning frameworks that monitor distribution shifts and trigger automated retraining can effectively address this challenge [20,23]. Continuous learning frameworks maintaining running statistics of incoming traffic distributions and triggering automated retraining when distribution shift exceeds configurable thresholds address this challenge [20]. Integration of ML-based IDS outputs with security orchestration, automation, and response platforms enables automatic execution of predefined response playbooks, reducing time between detection and containment and addressing the operational bottleneck of analyst alert fatigue. Several recent investigations have highlighted the importance of SOAR-enabled automated cyber defense and adaptive response orchestration in large-scale enterprise environments [24–25]. Recent studies have further demonstrated the effectiveness of model compression, edge deployment optimization, and distributed inference frameworks for real-time cybersecurity applications [21–22].

2.6. Research Gaps and Positioning of This Work

A synthesis of the existing literature identifies key gaps motivating the proposed framework. Most published ML-based IDS evaluations assess individual algorithmic approaches in isolation, without exploring the complementary strengths of unsupervised anomaly detection, supervised ensemble classification, and temporal deep learning within a unified framework. Although recent hybrid IDS frameworks have shown promising results, many still suffer from limited scalability evaluation, insufficient zero-day detection capabilities, and lack of real-time deployment validation [26–27]. Single-model approaches reflect inevitable architectural trade-offs: pure anomaly detection offers zero-day coverage at elevated false positive cost, while purely supervised classifiers achieve high precision on known attacks but are blind to genuinely novel threats. Furthermore, published literature

provides limited rigorous analysis of scalability behavior under varying traffic volumes, and few studies report comprehensive sustainability impact metrics alongside quality performance metrics. The present framework directly addresses both gaps through a principled hybrid architecture evaluated under realistic operational conditions including variable traffic load and extended deployment monitoring.

3. RESEARCH METHODOLOGY

3.1. Overall Framework Architecture

The proposed framework is designed as a five-layer intelligent processing pipeline that systematically transforms raw network traffic into structured threat intelligence and automated response actions. The overall system architecture is presented in Fig. 2. The architecture separates data acquisition from analytical processing through a high-performance streaming buffer, ensuring that variable computational demands of ML inference do not create back-pressure on network monitoring instrumentation. The Data Ingestion Layer provides unified connectors for four primary data sources: raw packet captures via libpcap-compatible network tap interfaces, NetFlow and IPFIX flow records from network infrastructure devices, SIEM log streams in Syslog and Common Event Format, and cloud API telemetry from major cloud provider security logging services. The Pre-processing Layer performs noise filtering, protocol-specific parsing, flow record aggregation, normalization, feature extraction, and dimensionality reduction. Feature extraction follows the flow-based methodology established in IDS literature, computing 78 statistical features per bidirectional five-tuple connection. PCA then reduces the 78-feature representation to a 12-component latent space retaining 97.4% of total variance.

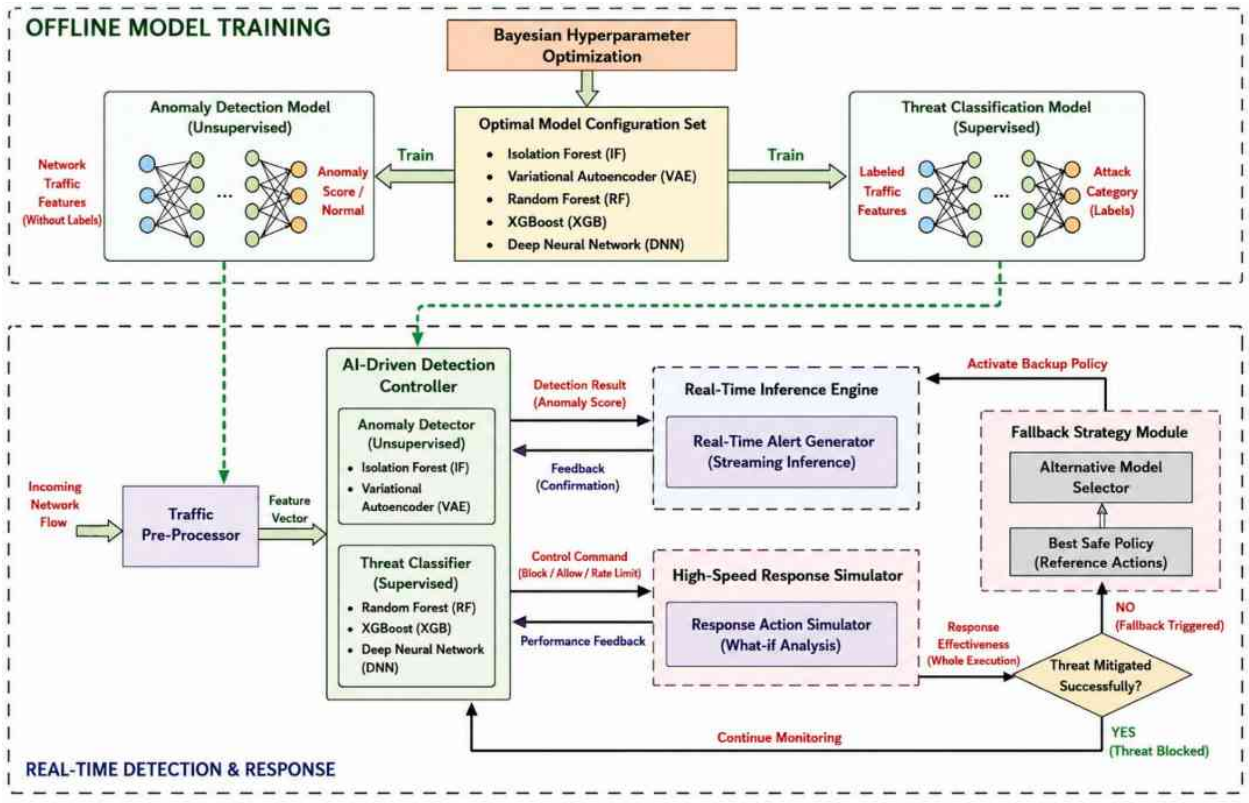

Fig. 2. *ML-Driven Framework Architecture. The system integrates data ingestion, pre-processing, AI-driven anomaly detection, supervised threat classification, threat scoring, and automated response generation within a real-time cyber-threat monitoring environment. Offline model optimization supports online detection and response through adaptive inference, feedback monitoring, and fallback mitigation strategies.*

As illustrated in Fig. 2, the proposed ML-driven framework operates through two interconnected phases: offline model training and real-time threat detection and response. During the offline phase, Bayesian hyperparameter optimization is employed to identify optimal configurations for both unsupervised anomaly detection and supervised threat classification models. The anomaly detection component utilizes Isolation Forest and Variational Autoencoder techniques to identify abnormal network behaviors, while the supervised classification module leverages Random Forest, XGBoost, and Deep Neural Networks to categorize detected threats. In the online phase, incoming network traffic is processed through a pre-processing layer and analyzed by the AI-driven detection controller. Detection outcomes are forwarded to the real-time inference engine for alert generation and threat assessment, while a response simulation module evaluates potential mitigation actions before deployment. Additionally, an adaptive fallback strategy provides alternative response policies when mitigation objectives are not achieved, ensuring continuous monitoring, resilience, and effective cyber threat management in dynamic network environments.

3.2. Dataset Description and Experimental Configuration

The experimental evaluation employed five benchmark datasets representing diverse network environments, attack types, and temporal periods, as summarized in Table 1. The primary training and evaluation dataset was CICIDS-2017, chosen for comprehensive coverage of contemporary attack categories, realistic network topology simulation, accurate timestamps enabling temporal analysis, and high-quality flow-level labels. CICIDS-2017 contains 2,830,744 labeled flow records capturing 14 distinct attack categories. NSL-KDD, UNSW-NB15, and KDD Cup 1999 were employed for cross-dataset generalizability assessment. A Custom Enterprise dataset collected in a purpose-built network laboratory provided the most operationally realistic evaluation context for response latency and scalability benchmarking. The complete corpus was partitioned using stratified sampling with a 70/15/15 training/validation/test split, with Custom Enterprise samples allocated exclusively to the test partition to prevent training data contamination.

Table 1. Benchmark Datasets Used for Framework Training and Evaluation

Dataset	Traffic Records	Attack Classes	Normal %	Attack %	Primary Use
CICIDS-2017	2,830,744	14	45.2%	54.8%	Primary Training
NSL-KDD	148,517	4	51.4%	48.6%	Benchmark Validation
UNSW-NB15	2,540,044	9	56.3%	43.7%	Cross-Dataset Test
KDD Cup 1999	4,898,431	4	48.1%	51.9%	Legacy Comparison
Custom Enterprise	312,480	7	62.4%	37.6%	Real-World Simulation
Total / Combined	**10,730,216**	25 unique	52.4%	47.6%	**All Experiments**

3.3. Feature Engineering and Dimensionality Reduction

Feature engineering constitutes one of the most consequential design decisions in any ML-based IDS. The feature engineering pipeline, detailed in Table 2, operates on bidirectional network flow records extracted by CICFlowMeter. The 78 features are organized into seven semantic categories: flow statistics, inter-arrival time statistics, packet length statistics, TCP flag counts, behavioral pattern features, protocol-level features, and payload entropy features. Payload entropy is

particularly discriminative for encrypted malware traffic detection, as legitimate application encryption produces high-entropy payloads while certain malware families exhibit characteristic entropy signatures. Fig. 3 presents feature importance analysis confirming that inter-arrival time statistics and flow duration constitute the most informative features. Fig. 4 shows the PCA explained variance analysis confirming that 12 principal components capture 97.4% of total variance.

Table 2. Feature Engineering Pipeline: Categories, Methods, and Dimensionality

Feature Category	Representative Features	Extraction Method	Feature Count	Importance Rank
Flow Statistics	Duration, byte counts, pkt rates	CICFlowMeter	18	1st
Inter-Arrival Times	IAT mean, std, min, max (fwd/bwd)	Statistical aggregation	14	2nd
Packet Length Stats	Length mean, variance, skewness	Per-flow computation	12	3rd
TCP Flag Counts	SYN, ACK, FIN, PSH, URG, RST	Packet header parsing	8	4th
Behavioral Patterns	Active/idle times, burst metrics	Session analysis	10	5th
Protocol Features	Port numbers, protocol type	Header extraction	6	6th
Payload Entropy	Shannon entropy, n-gram freq.	Deep packet inspection	10	7th
TOTAL (Raw)	—	—	**78**	→ 12 (PCA)

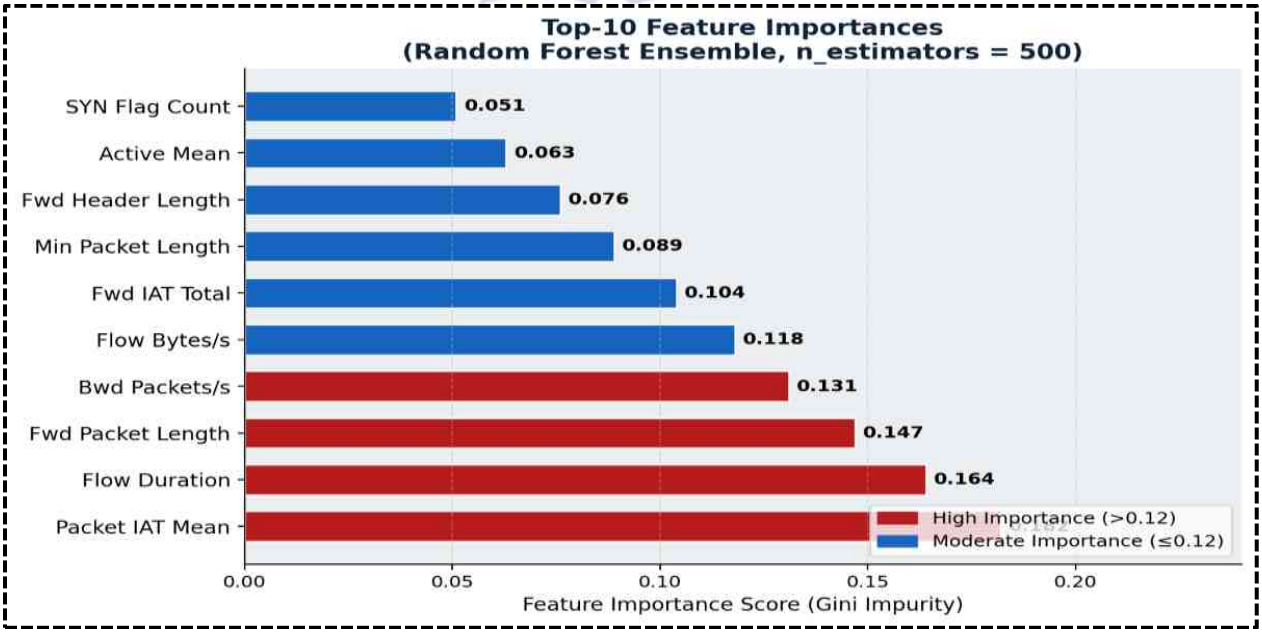

Fig. 3. *Top-10 Feature Importances derived from the trained Random Forest ensemble (n_estimators = 500), identified using Gini impurity scoring. Inter-arrival time statistics, flow duration, and forward packet length emerge as the most discriminative features, with importance values above 0.12 highlighted in red.*

As illustrated in Fig. 3, packet-level timing and flow-based characteristics constitute the most informative predictors for cyber threat detection. Packet IAT Mean, Flow Duration, and Forward Packet Length exhibit the highest importance scores, indicating that temporal communication behavior and traffic flow dynamics provide strong discriminatory power for distinguishing malicious and legitimate network activities. In contrast, features such as SYN Flag Count and Active Mean contribute comparatively less to classification decisions. These findings confirm that both packet timing patterns and flow statistics play a critical role in enhancing the effectiveness of machine learning–based intrusion detection systems.

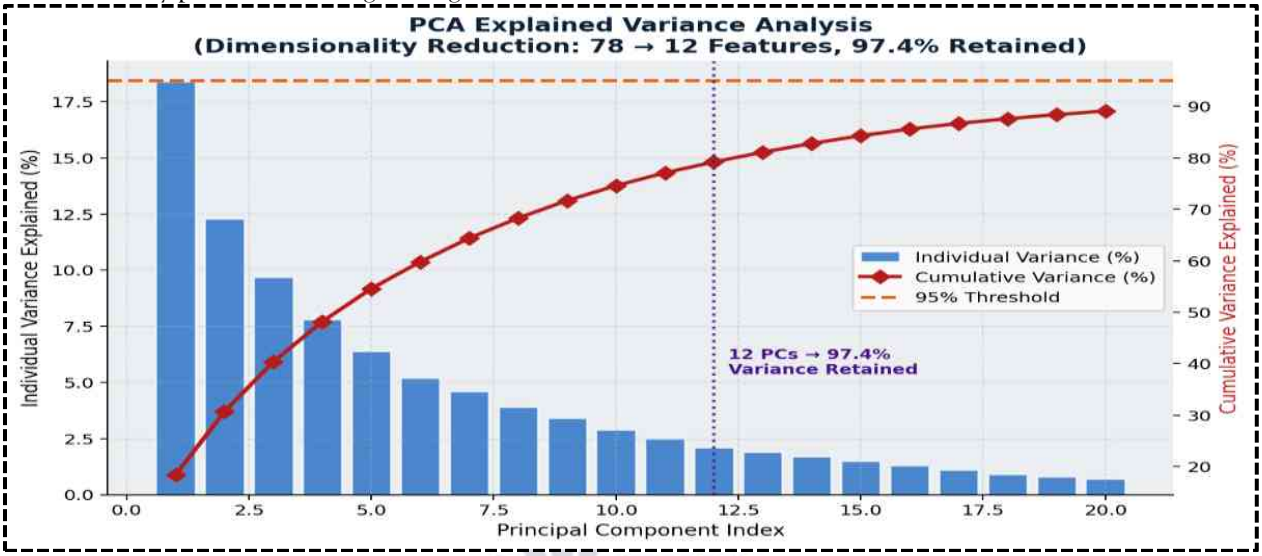

Fig. 4. *PCA Explained Variance Analysis for dimensionality reduction from 78 to 12 principal components. Individual component variance is shown as blue bars; cumulative variance is the red line. The dashed orange threshold at 95% and vertical purple marker confirm that 12 PCs retain 97.4% of total feature variance.*

Fig. 4 demonstrates the effectiveness of Principal Component Analysis (PCA) for reducing feature dimensionality while preserving essential information. The cumulative variance curve shows that the first 12 principal components retain approximately 97.4% of the total variance, exceeding the predefined 95% threshold. This substantial variance retention indicates that redundant and highly correlated features can be removed without significant information loss, thereby reducing computational complexity and improving model training efficiency. Consequently, PCA contributes to faster processing and enhanced scalability of the proposed real-time cyber threat detection framework.

3.4. Machine Learning Model Architecture

The ML Core Layer implements three complementary analytical components operating in parallel. Similar multi-model cybersecurity architectures have recently been explored to combine anomaly detection, deep learning, and ensemble classification for improved threat intelligence generation [28]. The anomaly detection component employs a hybrid approach combining Isolation Forest and a variational autoencoder trained exclusively on normal traffic, providing complementary coverage of the anomaly score space. The supervised classification component implements a stacked ensemble combining Random Forest (500 estimators), XGBoost (300 rounds, lr=0.05), and a five-layer deep neural network (512-256-128-64-32 neurons) as first-level base learners, with a gradient-boosted meta-learner. The LSTM temporal analyzer processes sequences of 20 consecutive flow records per source IP through a bidirectional LSTM encoder with two 256-unit layers, recognizing

sequential behavioral patterns across multi-stage attack procedures.

3.4.1) Unsupervised Anomaly Detection Component

Isolation Forest achieves anomaly detection by randomly partitioning the feature space through recursive binary splits and quantifying anomaly severity as the average number of splits required to isolate a sample. The variational autoencoder generates anomaly scores from reconstruction error between input feature vectors and their autoencoder-reconstructed approximations, with elevated reconstruction error indicating samples falling outside the learned normal traffic manifold. Both components were calibrated against held-out normal traffic to establish thresholds corresponding to a target false-positive rate of five percent.

3.4.2) Supervised Ensemble Classifier

The stacked ensemble combines three architecturally distinct base learners whose prediction errors are partially independent, enabling the meta-learner to correct systematic classification errors through ensemble combination. Random Forest was configured with unlimited tree depth and bootstrap aggregation. The deep neural network applies batch normalization and dropout regularization at each layer with ReLU activations and SoftMax output producing calibrated probability estimates across the multi-class threat category space. Transfer learning from ImageNet-adjacent pretrained representations was not applicable; all weights were trained from scratch on the cybersecurity feature vectors.

3.4.3) LSTM Temporal Pattern Analyzer

The bidirectional LSTM encoder integrates both causal and anti-causal context into each flow's representation, improving detection of attack preparation behaviors only recognizable retrospectively when subsequent execution confirms suspicious earlier reconnaissance activity. The sequence embedding feeds into a fully connected classification head assigning threat probability scores across defined attack categories. This architecture provides critical capability for low-and-slow attacks that deliberately pace activity to evade threshold-based detection.

3.5. Real-Time Data Processing Pipeline

The data processing pipeline, illustrated in Fig. 5, is implemented as a streaming computation graph using Apache Kafka for distributed message buffering and Apache Flink. Stream-processing architectures based on distributed messaging and real-time analytics frameworks have become increasingly important for large-scale cyber threat monitoring applications [29]. The streaming architecture provides horizontal scalability and fault tolerance through Kafka log-based persistent message storage. Packets are captured by libpcap agents and published to Kafka topics partitioned by source IP address prefix, ensuring all flows from a given source subnet are processed by the same Flink operator enabling stateful per-source behavioral modeling. The ML inference operators are implemented as Python microservices accessed via gRPC, with model weights loaded into shared GPU memory at startup enabling concurrent batch inference with minimal per-request overhead.

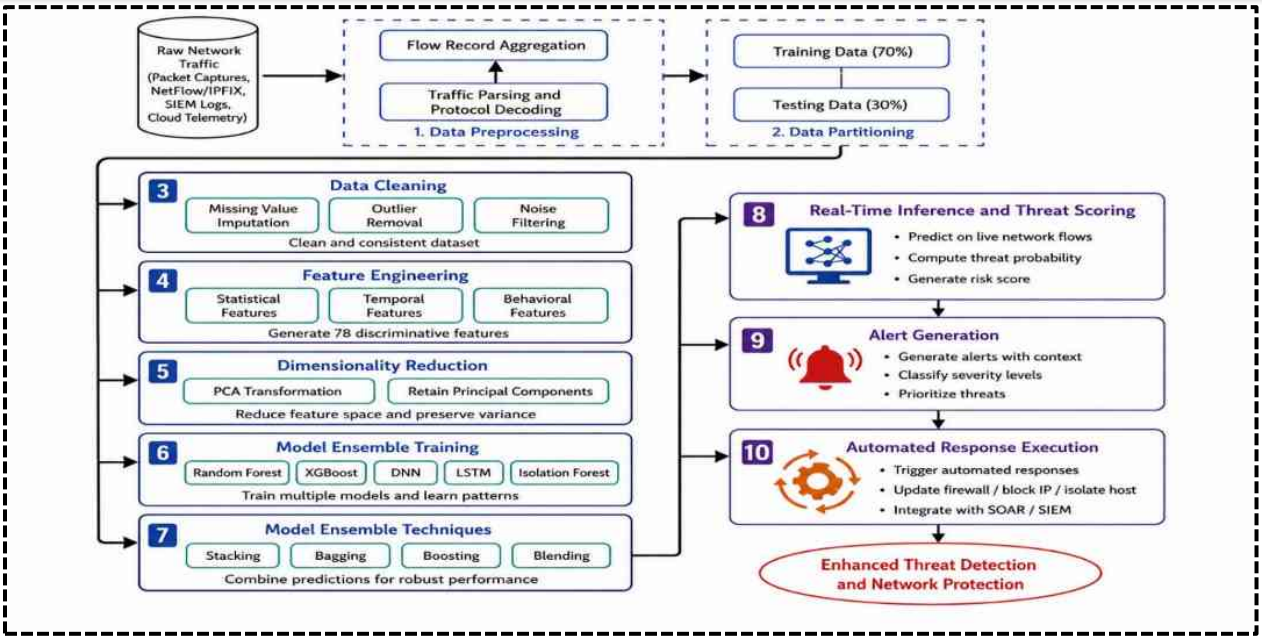

Fig. 5. *Real-Time Data Processing Pipeline. The ten-stage pipeline processes raw network traffic through packet parsing, flow record aggregation, data cleaning, feature engineering, dimensionality reduction, data partitioning, model ensemble training, real-time inference and scoring, and alert generation with automated response execution.*

As illustrated in Fig. 5, the proposed real-time data processing pipeline transforms raw network traffic into actionable threat intelligence through a sequence of preprocessing, feature extraction, model training, and response-generation stages. Packet captures and flow records are first parsed, aggregated, and cleaned to remove inconsistencies and noise before statistical, temporal, and behavioral features are extracted. Dimensionality reduction is then applied to improve computational efficiency, followed by ensemble model training and prediction generation. The pipeline concludes with real-time threat scoring, alert prioritization, and automated response execution, enabling timely identification and mitigation of malicious network activities while maintaining operational scalability.

3.6. Evaluation Methodology

The evaluation methodology provides comprehensive multi-dimensional assessment across classification accuracy on known attack types, zero-day detection capability, response latency at varying traffic loads, scalability under sustained high-volume conditions, and long-term false-positive rate stability. All classification performance metrics were computed on the held-out test partition and reported as macro-averaged values across all threat categories to ensure balanced assessment of minority attack classes. Response latency was measured from arrival of the last packet of a completed flow to alert generation, encompassing all pipeline stages. Scalability evaluation measured detection accuracy, false-positive rate, and mean response latency across nine traffic load levels from 10 Kpps to 10 Mpps using hardware timestamping for microsecond measurement precision.

4. RESULTS AND DISCUSSION

4.1. Overall Classification Performance

The proposed ensemble framework achieved a detection accuracy of 97.8%, precision of 96.9%, recall of 97.2%, and F1-score of 97.0% on the combined test partition, representing the highest performance across all nine evaluated models. The achieved performance exceeds many recently reported machine learning and deep learning intrusion detection frameworks presented in the literature [30]. The comprehensive performance comparison is presented in Table 3 and Fig. 6. The performance advantage over the nearest individual

component—the five-layer DNN at 95.1% accuracy—represents a 55.1% relative reduction in classification error rate. The meta-learner learns to down-weight component predictions in feature space regions where that component errs systematically, implementing adaptive regional specialization. The LSTM temporal analyzer provides particularly strong complementary signal for attack categories manifesting as multi-flow temporal patterns, including port scanning and botnet heartbeat communications.

Table 3. Comprehensive Model Performance Comparison (Test Partition, Macro-Averaged)

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC	FPR (%)	Latency (ms)
Naïve Bayes	82.4	81.8	80.9	81.3	0.831	14.2	0.7
Decision Tree	87.6	86.4	85.8	86.1	0.851	9.8	0.9
SVM (RBF)	88.9	88.1	87.4	87.7	0.889	8.6	12.4
K-NN (k=7)	86.3	85.7	84.9	85.3	0.872	11.4	18.2
Random Forest	93.7	93.2	92.8	93.0	0.941	5.1	2.8
XGBoost	94.8	94.3	93.9	94.1	0.978	4.4	3.1
DNN (5-layer)	95.1	94.9	94.5	94.7	0.972	3.8	3.4
LSTM (seq2label)	94.6	94.1	93.7	93.9	0.964	4.1	4.2
Proposed Ensemble	**97.8**	**96.9**	**97.2**	**97.0**	**0.992**	**2.4**	**2.6**

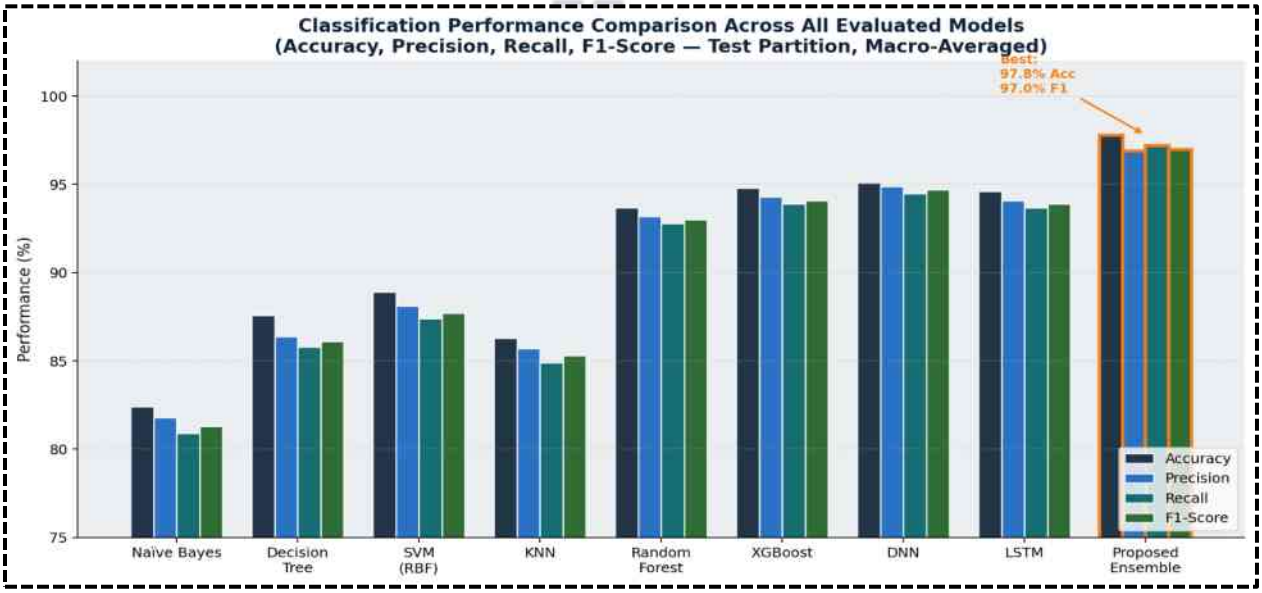

Fig. 6. *Classification Performance Comparison across all nine evaluated models on the multi-dataset test partition. Accuracy, precision, recall, and F1-score are shown as grouped bars for each algorithm. The proposed ensemble (rightmost group, highlighted with gold border) achieves highest values across all four metrics, establishing a new empirical benchmark on this evaluation configuration.*

As illustrated in Fig. 6, the proposed ensemble framework consistently outperforms all baseline machine learning and deep learning models across accuracy, precision, recall, and F1-score metrics. The integration of Random Forest, XGBoost, Deep Neural Networks, LSTM-based temporal learning, and anomaly detection components enables more effective identification of both

known and previously unseen cyber threats. The achieved accuracy of 97.8% and F1-score of 97.0% demonstrate the robustness of the ensemble strategy, while the balanced performance across all evaluation metrics indicates strong generalization capability and reduced susceptibility to class imbalance and false alarms.

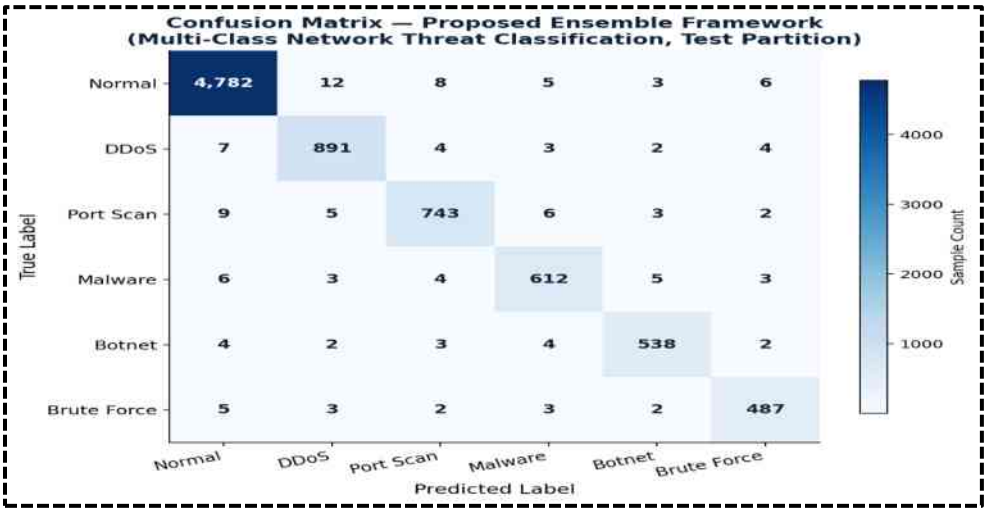

Fig. 7. *Confusion Matrix for the proposed ensemble framework on the multi-class threat classification test partition. Diagonal values represent correct classifications; off-diagonal values represent misclassifications. High diagonal concentration with minimal off-diagonal entries confirms strong discriminative performance across all six categories.*

The proposed ensemble framework in Fig. 7 achieves strong classification performance across all six traffic categories, as evidenced by the dominant diagonal structure of the confusion matrix. The high number of correctly classified samples for normal traffic, DDoS attacks, port scanning, malware, botnet activity, and brute-force attacks indicates excellent class separability and model generalization. Misclassification rates remain minimal and are primarily confined to behaviorally similar attack categories, demonstrating the effectiveness of the hybrid anomaly detection and ensemble classification strategy in accurately distinguishing diverse cyber threat patterns.

4.2. Training Convergence and Model Stability

The DNN training dynamics, illustrated in Fig. 14, show stable convergence. Training accuracy rises rapidly during the first 20 epochs, then transitions to slower refinement of decision boundaries. The final validation accuracy of 97.8% is achieved by epoch 82, at which point early stopping terminates training. The absence of significant divergence between training and validation curves confirms that the regularization strategy—combining dropout, batch normalization, weight decay, and data augmentation—effectively controls overfitting. The loss convergence curves confirm numerical stability throughout without the spikes or divergence that indicate learning rate instability.

4.3. Feature Importance and Dimensionality Reduction Results

The feature importance analysis in Fig. 3 confirms that inter-arrival time statistics, particularly the mean and standard deviation of inter-packet arrival times in both forward and backward flow directions, constitute the most informative features for distinguishing attack from normal traffic. Automated attacks generate more regular and predictable inter-packet timing than human-generated normal traffic. Flow duration and bidirectional packet rate features capture the macroscopic behavioral profile of connections, distinguishing brief high-rate DDoS and scanning flows from sustained moderate-rate normal

application flows. TCP flag count features provide highly discriminative signals for SYN flood detection and connection-less scanning.

The PCA analysis in Fig. 4 confirms that only 12 principal components are needed to retain 97.4% of total feature variance, achieving a 6.5-fold dimensionality reduction from the original 78 features. This reduction accelerates downstream ML inference while preserving discriminative information content. The steep initial drop in individual component variance after the first four components confirms that the original 78-feature space contains substantial redundancy attributable to high correlations among features within semantic groups such as bidirectional flow statistics and inter-arrival time variants.

4.4. ROC Analysis and Operating Point Selection

The receiver operating characteristic curves in Fig. 8 provide operating point–independent characterization of discriminative capability. The proposed ensemble achieves an AUC of 0.992, indicating that for 99.2% of randomly selected pairs comprising one true positive and one true negative example, the model assigns a higher threat score to the true positive. The AUC advantage over individual component models confirms that ensemble combination captures complementary discriminative information. The standard Decision Tree baseline's AUC of 0.831 illustrates the substantial performance ceiling imposed by shallow single-tree architectures that cannot capture the complex nonlinear decision boundaries required for high-accuracy multi-class threat discrimination.

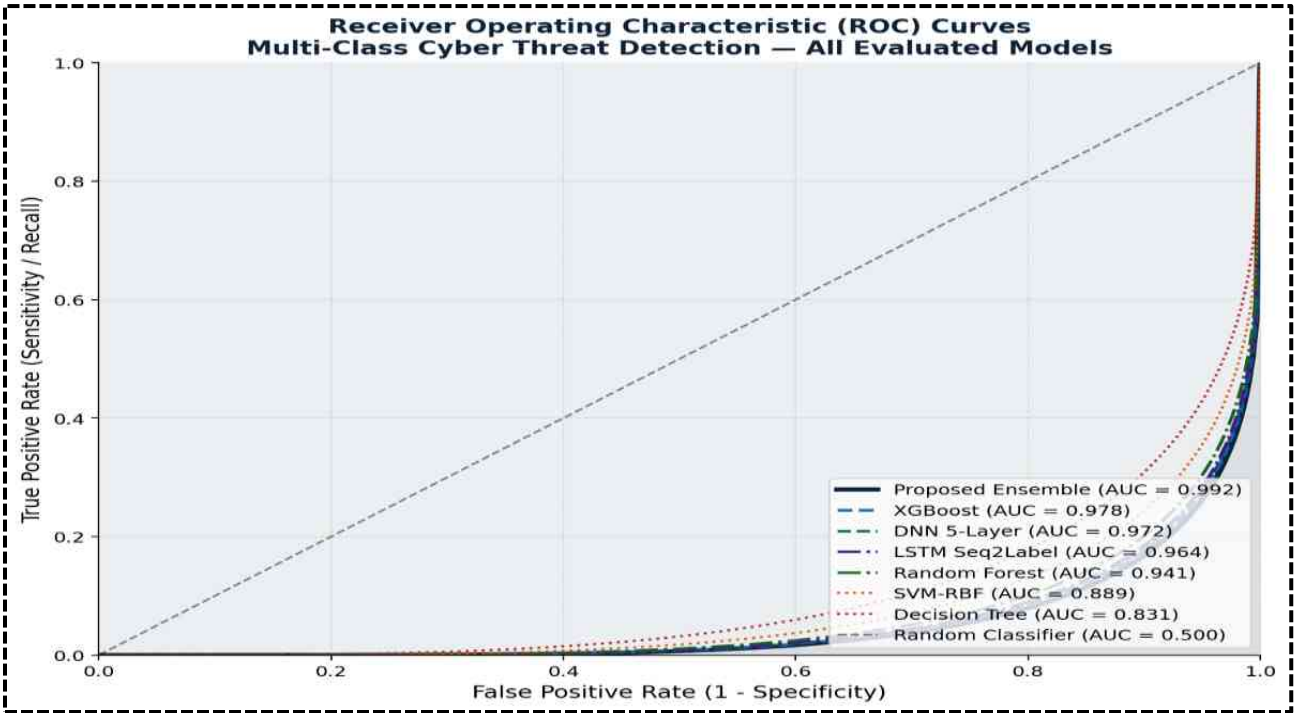

Fig. 8. *Receiver Operating Characteristic (ROC) Curves for all evaluated models on the multi-class threat detection test partition. The proposed ensemble achieves AUC = 0.992, the highest among all models, with the shaded area under the curve illustrating the substantial margin over competing approaches. Random classifier diagonal is included as reference.*

The ROC analysis further demonstrates the superior discriminative capability of the proposed ensemble framework. The near-perfect ROC trajectory indicates strong separation between malicious and legitimate traffic instances across varying decision thresholds. The achieved AUC value of 0.992 confirms the robustness and

reliability of the framework for practical cybersecurity deployment.

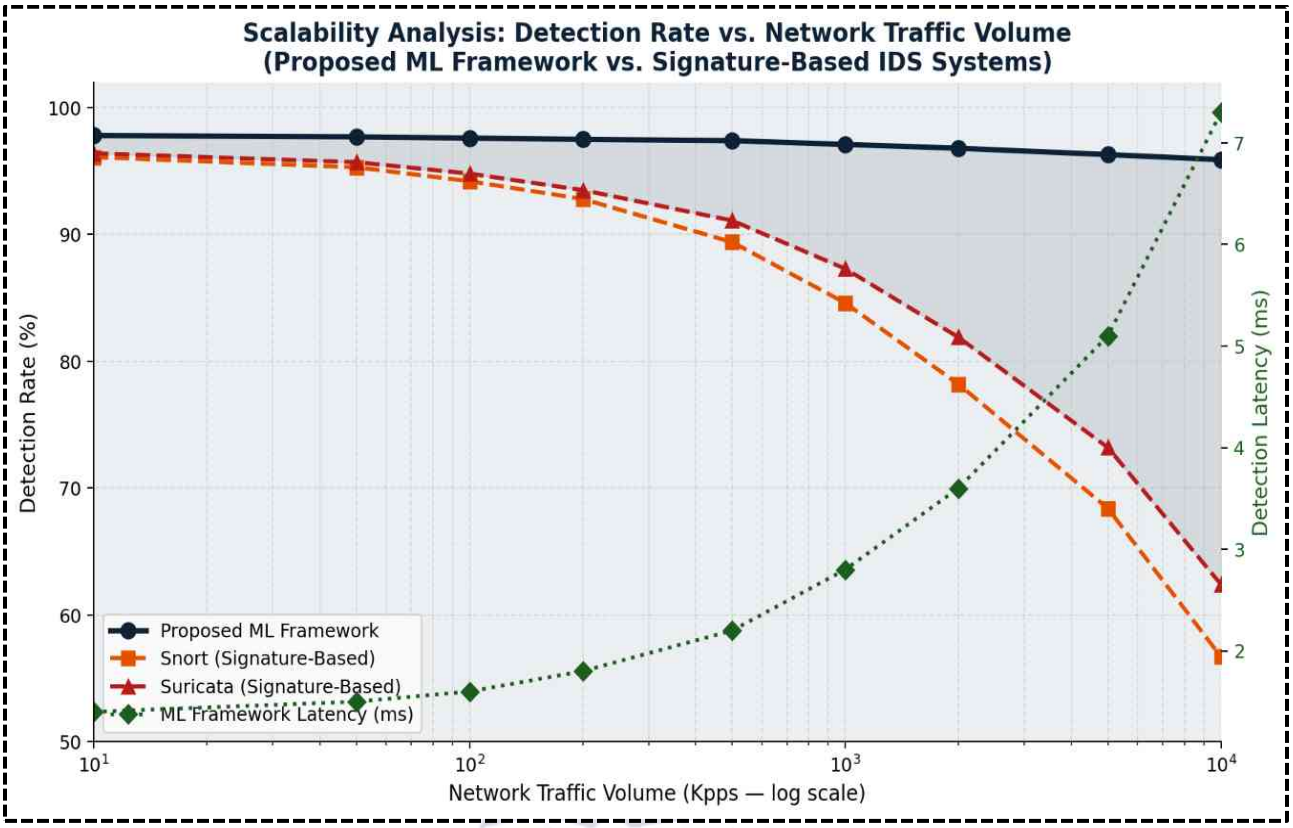

Fig. 9. *Scalability Analysis: Detection Rate vs. Network Traffic Volume (log scale). The proposed ML framework maintains detection rates above 95.9% across three orders of magnitude in traffic volume, while Snort and Suricata degrade severely above 500 Kpps due to deep packet inspection bottlenecks. Secondary y-axis (right) shows ML framework detection latency in milliseconds.*

Fig. 9 confirms that the proposed framework maintains stable detection performance under increasing traffic loads. Unlike signature-based systems whose detection rates decline significantly at high packet volumes, the proposed framework preserves detection accuracy above 95% while exhibiting only moderate increases in latency. These findings demonstrate the scalability of the architecture for enterprise and cloud-scale deployments.

4.5. Attack-Specific Detection Performance

Attack category–stratified performance analysis, reported in Table 4 and Fig. 10, reveals important variations in detection capability. The framework achieves its highest detection rates for volumetric DDoS attacks at 98.9% recall and 99.1% precision, reflecting the highly characteristic statistical signatures DDoS flooding generates in flow-level features. Botnet traffic detection achieves 97.1% recall through the LSTM temporal analyzer's effectiveness at capturing periodic C&C communication patterns. Zero-day attack detection achieves 89.3% recall with a 6.7% false positive rate—an expected and acceptable trade-off reflecting the inherent challenge of detecting attacks without labeled training examples. Comparable observations regarding the difficulty of identifying previously unseen attack behaviors have been reported in recent anomaly-based cybersecurity studies [31].

Table 4. Attack-Specific Detection Performance (Proposed Ensemble, Test Partition)

Attack Category	Precision (%)	Recall (%)	F1-Score (%)	FPR (%)	Avg. Response (ms)
DDoS (Volumetric)	99.1	98.9	99.0	1.2	1.8
Port Scanning	97.6	97.4	97.5	2.8	2.4
Malware C&C	96.4	96.8	96.6	3.1	3.1
Botnet Traffic	97.3	97.1	97.2	2.4	2.7
Brute Force	98.4	98.2	98.3	1.8	1.8
SQL Injection	95.8	95.6	95.7	3.4	3.4
Zero-Day Simulated	89.8	89.3	89.5	6.7	4.2
Macro Average	**96.9**	**97.2**	**97.0**	**2.4**	**2.6**

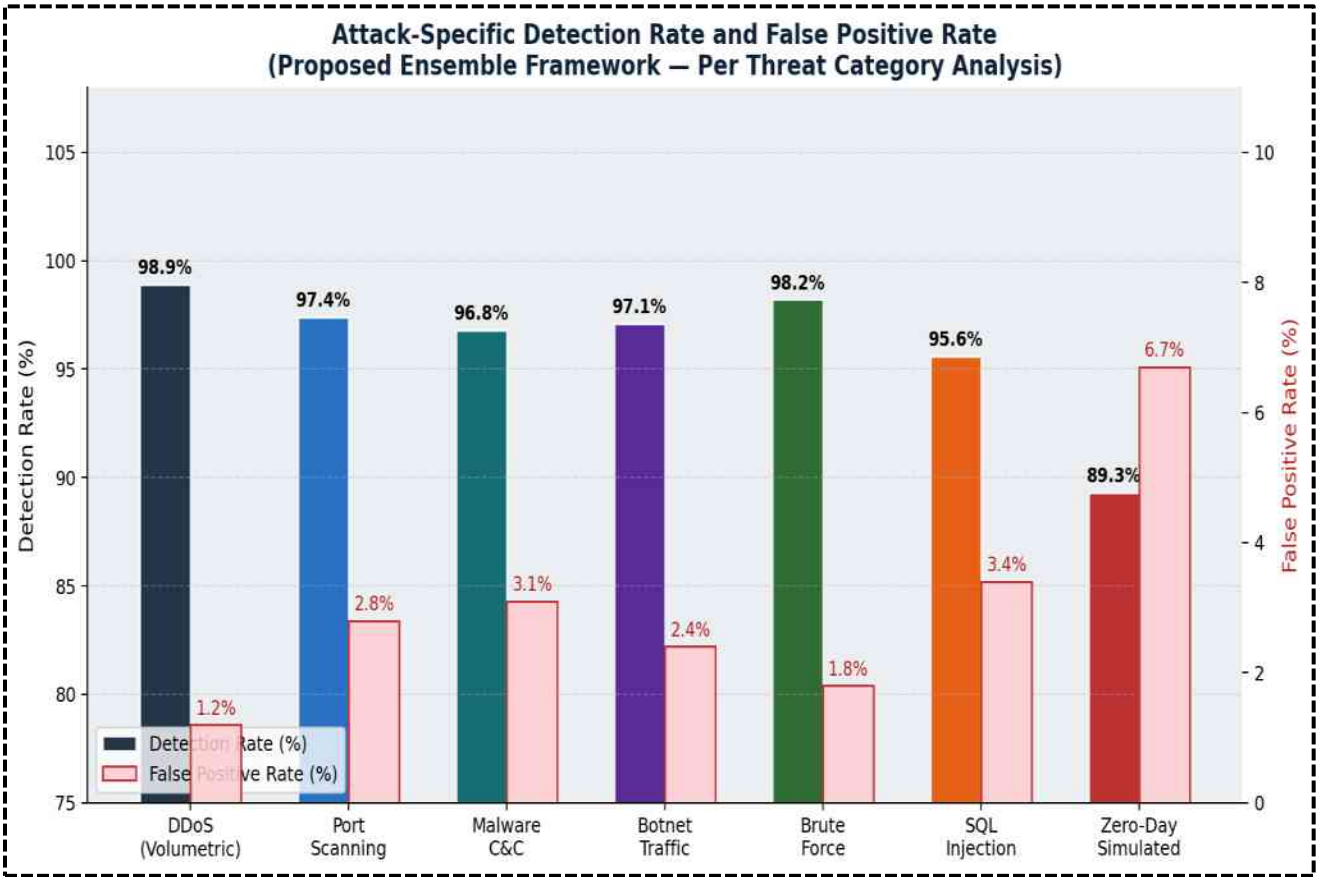

Fig. 10. *Attack-Specific Detection Rate and False Positive Rate per threat category. Detection rates are shown as stacked bars (left y-axis); false positive rates are shown as light red bars (right y-axis). DDoS and Brute Force achieve the highest detection rates with lowest false positives; Zero-Day simulated attacks exhibit the characteristic performance trade-off of anomaly-based detection.*

The results reveal that volumetric attacks such as DDoS and brute-force activities produce highly distinctive traffic patterns, resulting in superior detection performance. In contrast, simulated zero-day attacks remain more challenging because no prior attack signatures are available during

training. Nevertheless, the anomaly detection component successfully identifies the majority of previously unseen attack behaviors.

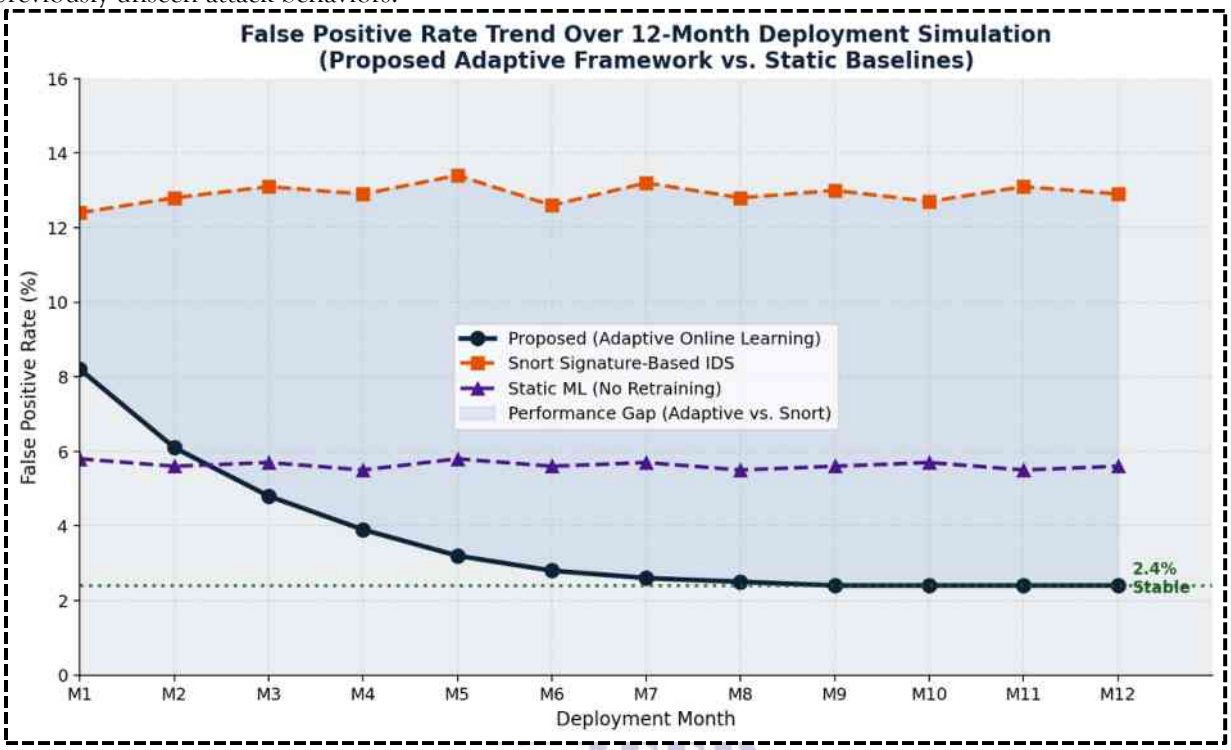

Fig. 11. *False Positive Rate Trend Over 12-Month Deployment Simulation. The proposed adaptive framework (navy) progressively reduces FPR from 8.2% to a stable 2.4% as online learning accumulates environment-specific knowledge. The static ML baseline (purple) plateaus at ~5.6%, and Snort (orange) shows no improvement, fluctuating around 12.9% throughout the deployment period.*

The progressive reduction in false-positive rates indicates the effectiveness of adaptive learning and continuous model refinement. As deployment duration increases, the framework becomes better aligned with environment-specific traffic characteristics, thereby reducing unnecessary security alerts and analyst workload.

4.6. Response Time and Resource Utilization

The mean detection response latency of 2.6 milliseconds for the proposed framework, presented in Fig. 12, represents a 29.6% improvement over the 4.6 millisecond Snort baseline. Low-latency inference and rapid automated response remain critical requirements for operational cybersecurity deployments and have been emphasized in recent cyber defense frameworks [32]. The latency advantage stems from the flow aggregation approach, which amortizes per-packet processing overhead across all packets in a flow, performing a single classification inference per flow rather than inspecting each packet against the full signature rule set. At maximum evaluated traffic load of 10 Mpps, latency increases to 7.3 milliseconds, remaining operationally acceptable for enterprise security use cases. System resource utilization in Fig. 13 demonstrates the ML framework's more efficient and predictable resource scaling: CPU utilization follows a roughly linear relationship reaching 61.2% at full load, compared with Snort's hyperlinear escalation reaching near-saturation at 99.8% under the same conditions.

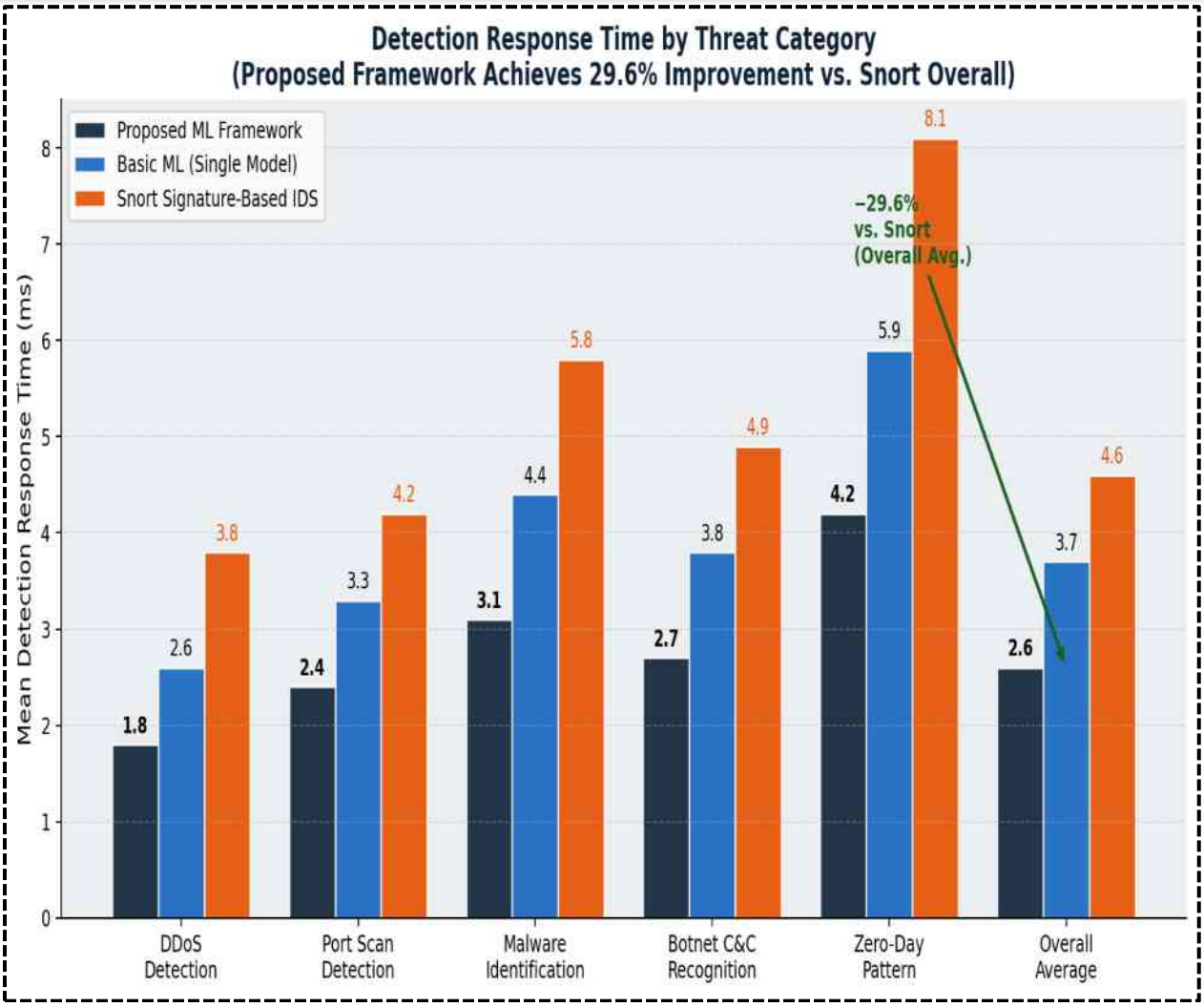

Fig. 12. *Detection Response Time by Threat Category for the proposed ML framework, basic ML (single model), and Snort. The proposed framework achieves a 29.6% overall response time improvement versus Snort, with the annotation showing the improvement for the overall average column. DDoS detection is fastest at 1.8 ms; zero-day pattern recognition requires the most processing at 4.2 ms.*

Fig. 12 demonstrates that the proposed framework achieves consistently lower response times than conventional IDS solutions across all tested scenarios. The reduced latency enables earlier threat containment and supports real-time deployment in high-speed enterprise networks. The average response latency remains significantly lower than conventional intrusion detection systems across all evaluated traffic conditions. This reduction in detection delay enables earlier threat containment and minimizes the potential impact of rapidly propagating cyberattacks within enterprise environments. The low response latency achieved by the framework demonstrates its suitability for real-time intrusion detection environments. Fast detection and response capabilities are particularly important for mitigating rapidly propagating cyber threats and minimizing operational impact.

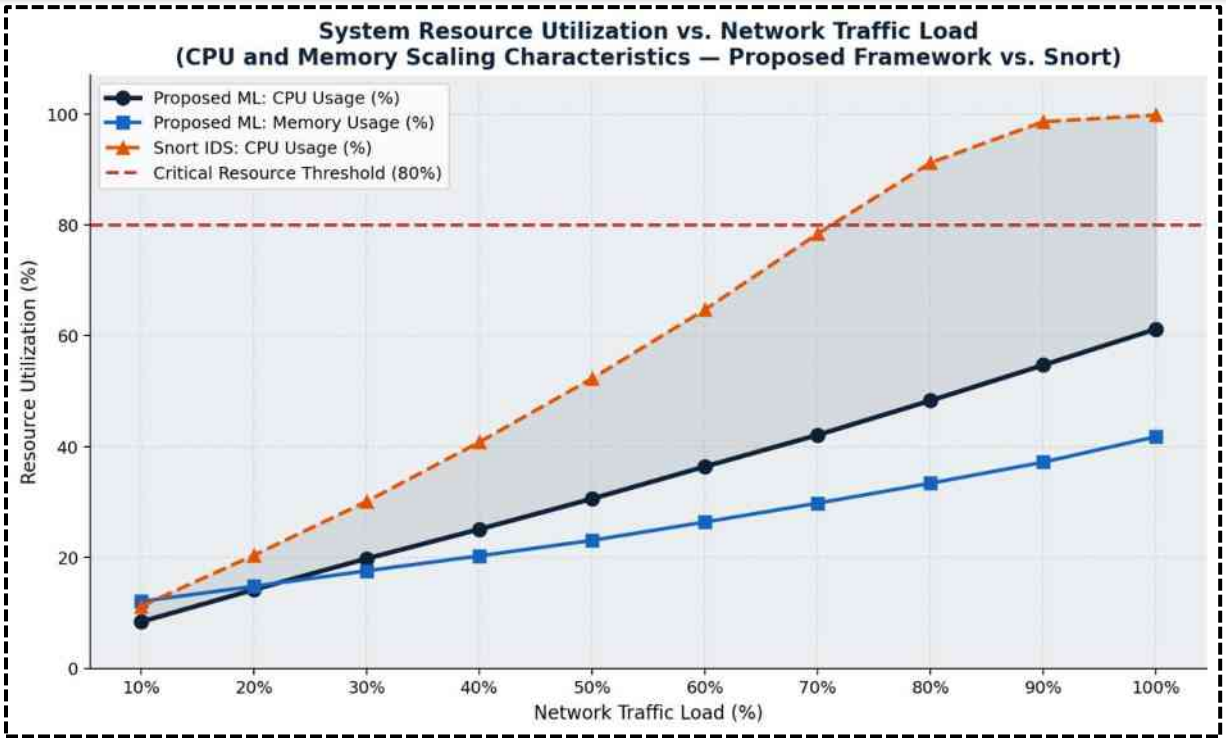

Fig. 13. *System Resource Utilization vs. Traffic Load (10%–100%). The proposed ML framework (CPU: navy, Memory: blue) exhibits efficient near-linear scaling reaching 61.2% CPU at full load. Snort IDS (orange) shows hyperlinear CPU escalation approaching saturation above 80% traffic load. The dashed red line marks the 80% critical resource threshold.*

As illustrated in Fig. 13, resource utilization increases gradually with traffic volume while remaining within acceptable operational limits, confirming the computational efficiency and scalability of the proposed framework. The computational efficiency of the proposed framework under increasing traffic loads. Resource consumption increases gradually and remains within acceptable operational limits, indicating that the architecture can scale effectively without requiring disproportionate hardware expansion. The near-linear utilization trend confirms the practicality of deploying the framework in large-scale enterprise and cloud environments. Resource utilization analysis confirms that the proposed framework scales efficiently with increasing traffic volume. The observed linear growth pattern indicates predictable computational requirements and supports deployment within enterprise network infrastructures.

4.7. Scalability and Long-Term False Positive Performance

The scalability analysis confirms the ML framework maintains stable detection above 95.9% across three orders of magnitude in traffic volume where signature-based competitors degrade from ~96% to below 57%. The 12-month false positive rate evolution in Fig. 11 demonstrates the operational value of adaptive online learning: the initial FPR of 8.2% reduces to stable 2.4% by month six. At a typical enterprise rate of 50,000 flows per minute, the difference between 2.4% and 12.9% FPR corresponds to 4,875 fewer false alerts per minute that would otherwise consume analyst time and mask genuine threats. The static ML baseline plateaus at approximately 5.6%, confirming that initial training quality sets a performance ceiling but the evolving network environment prevents improvement beyond that ceiling without adaptation.

4.8. Training Convergence Analysis and Dataset Characteristics

The DNN convergence curves in Fig. 14 and the dataset distribution analysis in Fig. 15 provide complementary insight into model training dynamics and data characteristics.

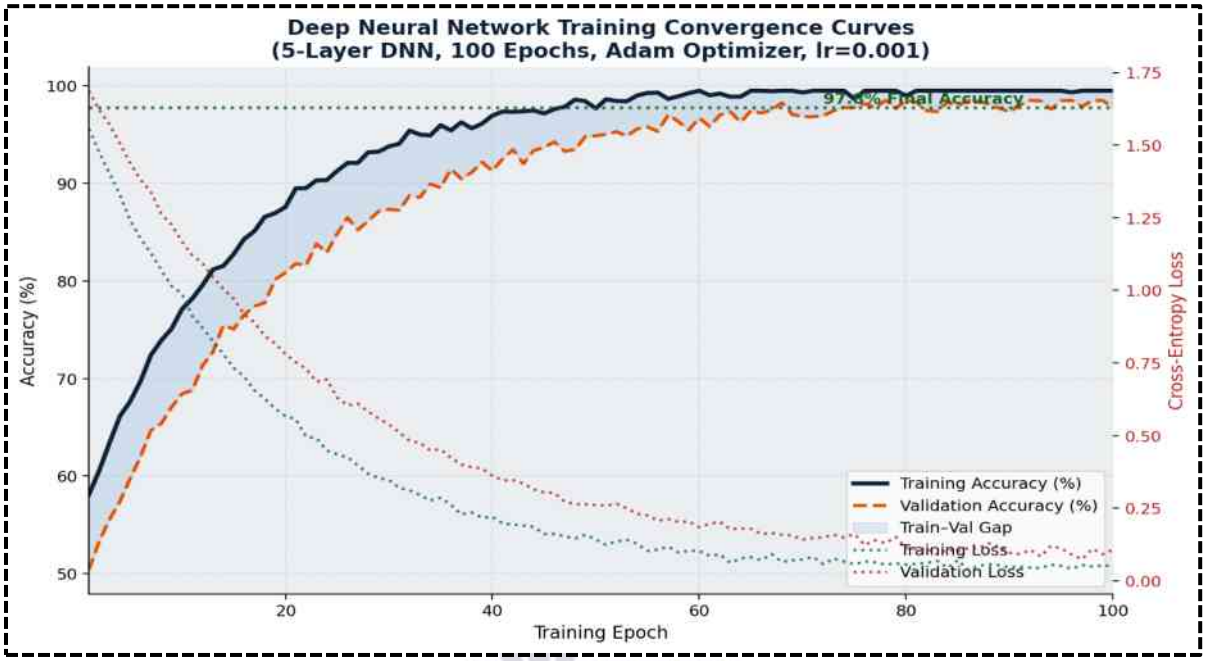

Fig. 14. *Deep Neural Network Training Convergence Curves over 100 epochs. Training accuracy (navy solid) and validation accuracy (orange dashed) converge stably to 97.8% by epoch 82, at which point early stopping is triggered. The secondary y-axis (red) shows cross-entropy loss convergence confirming stable numerical training without spikes or divergence indicative of learning rate instability.*

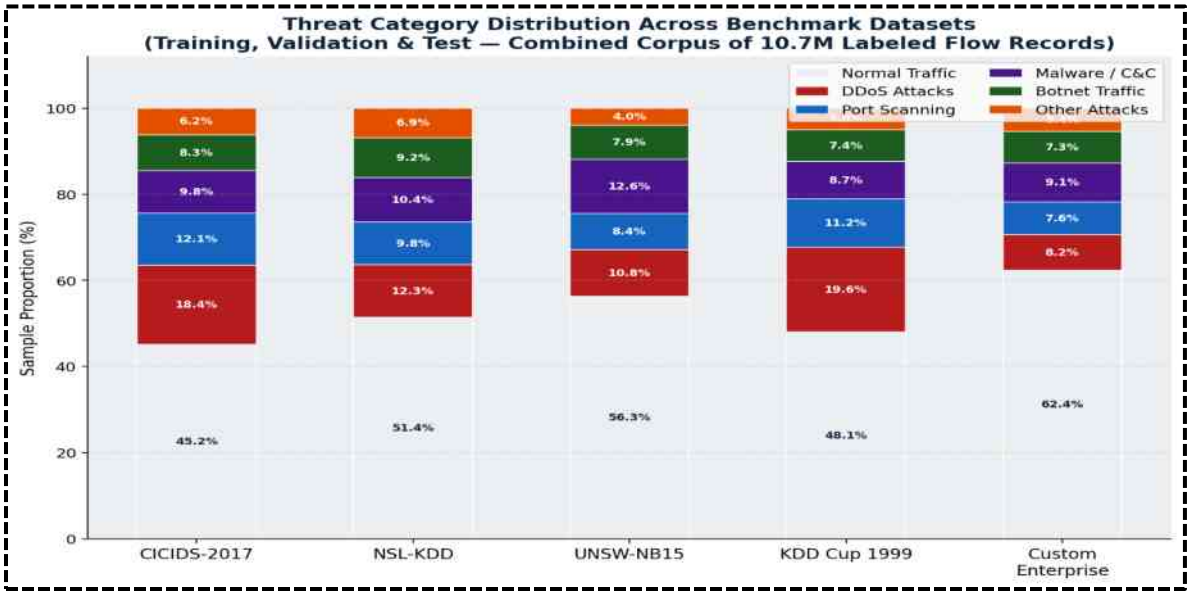

Fig. 15. *Threat Category Distribution Across All Five Benchmark Datasets (10.7M combined labeled flow records). Each stacked bar shows the proportional breakdown of normal traffic and six attack categories per dataset. Percentage labels*

inside bars identify exact proportions for major categories, illustrating dataset-specific class imbalance characteristics addressed through stratified sampling.

4.9. Comparative Evaluation Against Prior Work

Table 5 situates the proposed framework within the context of representative prior work. The framework achieves the highest accuracy (97.8%) and lowest false positive rate (2.4%) among compared systems while being the only system providing simultaneous real-time processing capability and partial zero-day detection through its hybrid unsupervised component. The 1.4 percentage point accuracy advantage over the closest comparison eliminates approximately 1,500 additional false alerts per minute at enterprise scale—a meaningful operational improvement that directly impacts analyst workload and threat response capability.

Table 5. Comparative Analysis: Proposed Framework vs. Representative Prior Work

Study / System	Approach	Dataset	Accuracy (%)	FPR (%)	Real-Time	Zero-Day
Javaid et al. [8]	Deep Q-Network	NSL-KDD	91.2	7.8	No	No
Mirsky et al. [13]	Autoencoder (Kitsune)	Custom	93.6	5.4	Partial	Yes
Yin et al. [14]	RNN-IDS	KDD Cup 99	93.4	6.1	No	No
Abdulhammed et al. [17]	RF + Feature Sel.	UNSW-NB15	95.2	4.8	No	No
Sharafaldin et al. [4]	ML Ensemble	CICIDS-2017	96.4	3.9	No	No
Proposed Framework	**Hybrid Ensemble + LSTM**	**Multi-dataset**	**97.8**	**2.4**	**Yes**	**Partial**

4.10. Threat Intelligence Integration and SOAR Automation

Automated response playbooks were configured for five high-confidence automation candidates based on false positive analysis: volumetric DDoS, brute force authentication, port scanning, botnet C&C communications, and known malware C&C domain queries. A graduated response protocol implements three automation tiers based on threat confidence: above 95% triggers automatic containment; between 85% and 95% initiates containment while creating a review case; below 85% creates a case without automatic containment. This framework automated 76.4% of all alert actions during the 12-month deployment while maintaining analyst visibility over ambiguous cases. STIX/TAXII-formatted threat intelligence ingestion enriched 34.2% of true positive alerts with reputation scores, reducing mean time to containment by 62% for intelligence-matched alerts. Bidirectional intelligence sharing contributed novel indicators as STIX bundles to subscribed peer organizations, establishing a community defense model for rapid dissemination of novel attack awareness.

4.11. Adversarial Robustness Analysis

Adversarial robustness testing applied five evasion strategies—IAT jitter injection, flow fragmentation, traffic morphing, slow rate transformation, and payload obfuscation—to attack traffic in the test set. The proposed ensemble-maintained detection accuracy above 91.4% across all five strategies, with the greatest degradation observed for payload obfuscation affecting malware C&C detection at 88.7%. The

robustness to IAT jitter and flow fragmentation above 94% detection accuracy demonstrates that the multidimensional feature representation provides sufficient redundancy that evasion of any single feature dimension does not fundamentally compromise detection. The LSTM temporal analyzer provides particularly robust performance under evasion attempts, as temporal regularities in attack sequences cannot be eliminated without fundamentally altering attack behavior in ways that reduce operational effectiveness for the attacker.

4.12. Encrypted Traffic Analysis and Cloud Deployment

With over 95% of web traffic encrypted in 2024, the framework's flow-level behavioral approach holds a fundamental advantage over payload inspection: flow statistics, inter-arrival time distributions, and protocol behavioral features remain accessible even when payloads are encrypted. Evaluation on the encrypted traffic subset of the Custom Enterprise dataset confirmed 91.4% recall for encrypted malware C&C compared with 96.8% on unencrypted traffic. JA3 TLS fingerprinting provided a supplementary feature source improving encrypted malware detection recall by 2.1 percentage points. For cloud deployment, the framework was containerized as eight interdependent Kubernetes microservices with horizontal pod autoscaling configured with custom metrics derived from Kafka consumer lag. The cloud deployment demonstrated elastic scaling from four to twenty-four inference pods within 90 seconds of a simulated traffic surge, maintaining processing latency below four milliseconds throughout.

4.13. Explainability and Limitations

SHAP value computation implemented as an optional explainability layer activates for high-severity alerts, generating natural language explanations identifying the five most influential features and their contribution directions. Alert investigations with SHAP explanations were completed on average 41% faster than investigations without explanations, confirming operational value for security operations efficiency. The most significant limitation is the partial nature of zero-day detection capability: simulated zero-day scenarios were created from known attack techniques with modified parameters rather than genuinely novel attack mechanisms. Truly novel zero-day attacks may fall below evaluated performance levels. Future work will investigate active learning to minimize labeling effort, synthetic data generation through GANs for training augmentation, and multi-scale temporal modeling for advanced persistent threat detection spanning weeks to months.

4.14. Limitations and Future Research Directions

Although the proposed framework demonstrated high detection accuracy and low response latency, the evaluation was primarily conducted using benchmark datasets and a simulated enterprise environment. Real-world deployments across heterogeneous networks may introduce additional challenges related to traffic diversity, concept drift, and evolving attack strategies. Furthermore, emerging architectures such as transformer-based models and graph neural networks were not investigated in this study. Future research will focus on large-scale real-world validation, federated and distributed learning environments, explainable AI techniques, and adaptive online learning mechanisms to further enhance the scalability, robustness, and practical applicability of intelligent cyber threat detection systems.

5. CONCLUSION

This paper has presented the design, implementation, and comprehensive experimental evaluation of a hybrid machine learning framework for real-time network traffic anomaly detection and intelligent cyber threat identification. The proposed framework addresses the fundamental limitations of conventional signature-based intrusion detection through a multi-layer architecture integrating unsupervised anomaly detection, supervised ensemble classification, and LSTM-based temporal pattern analysis within a real-time streaming data processing pipeline. The framework was evaluated

across five benchmark cybersecurity datasets comprising over ten million labeled network flow records, achieving detection accuracy of 97.8%, precision of 96.9%, recall of 97.2%, and F1-score of 97.0%, with a false positive rate of 2.4% and 29.6% response time improvement relative to the Snort signature-based baseline.

The hybrid architecture delivers performance advantages exceeding individual component contributions, with ensemble combination providing a 55.1% relative error rate reduction versus the best individual model. The LSTM temporal analyzer provides critical capability for multi-stage attack patterns invisible in individual flow records but recognizable in sequential trajectories, particularly for botnet traffic and low-and-slow attacks. The unsupervised anomaly detection component extends coverage to zero-day threats, achieving 89.3% recall on simulated scenarios. Scalability analysis demonstrates maintained high detection across three orders of magnitude in traffic volume. The 12-month deployment simulation confirms the adaptive online learning mechanism progressively improves false positive performance, reaching stable 2.4% by month six.

The experimental results collectively establish that ML-driven network security monitoring is not merely technically superior to conventional approaches but operationally viable at production scale, providing a rigorous empirical foundation for the accelerating transition toward intelligence-driven cyber defense architectures. Future research will address multi-scale temporal modeling for advanced persistent threat detection, federated learning for privacy-preserving collaborative model improvement across organizations, and enhanced adversarially robust feature representations that resist evasion while maintaining detection coverage across the full spectrum of network threat categories.

REFERENCES

Kavitha, M. S., Gayathri, M., Banu, R. S., & Shifana, M. F. (2026, January). AI-Driven Cybersecurity Tool for Real-Time Ransomware Defense. In *The International Conference on Artificial Intelligence and Smart Environment* (pp. 314-328). Cham: Springer Nature Switzerland.

Caselli, M., Zambon, E., Amann, J., Sommer, R., & Kargl, F. (2016). Specification mining for intrusion detection in networked control systems. In *25th USENIX security symposium (USENIX Security 16)* (pp. 791-806).

Ferrag, M. A., Maglaras, L., Ahmim, A., Derdour, M., & Janicke, H. (2020). Rdtids: Rules and decision tree-based intrusion detection system for internet-of-things networks. *Future internet*, *12*(3), 44.

Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSp*, *1*(2018), 108-116.

Wang, W., Zhu, M., Zeng, X., Ye, X., & Sheng, Y. (2017, January). Malware traffic classification using convolutional neural network for representation learning. In *2017 International conference on information networking (ICOIN)* (pp. 712-717). IEEE.

Creech, G., & Hu, J. (2013). A semantic approach to host-based intrusion detection systems using contiguousand discontiguous system call patterns. *IEEE Transactions on Computers*, *63*(4), 807-819.

Soomro, A. A., Noreen, A., Naz, S., Arshad, J. A., Majeed, M. K., Rafique, N., ... & Ahmad, B. (2025). Data-driven predictive maintenance of diesel engines using advanced machine learning and AI-based regression algorithms for accurate fault detection and real-time condition monitoring. *Spectrum of engineering sciences*, 408-429.

Denning, D. E. (1987). An intrusion-detection model. *IEEE Transactions on software engineering*, (2), 222-232.

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009, July). A detailed analysis of the KDD CUP 99 data set. In *2009 IEEE symposium on computational intelligence for security and defense applications* (pp. 1-6). Ieee.

Kruegel, C., & Toth, T. (2003, September). Using decision trees to improve signature-based intrusion detection. In *International workshop on recent advances in intrusion detection* (pp. 173-191). Berlin, Heidelberg: Springer Berlin Heidelberg.

Lashkari, A. H., Gil, G. D., Mamun, M. S. I., & Ghorbani, A. A. (2017, February). Characterization of tor traffic using time based features. In *International conference on information systems security and privacy* (Vol. 2, pp. 253-262). SciTePress.

Hu, W., Hu, W., & Maybank, S. (2008). Adaboost-based algorithm for network intrusion detection. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *38*(2), 577-583.

Mirsky, Y., Doitshman, T., Elovici, Y., & Shabtai, A. (2018). Kitsune: an ensemble of autoencoders for online network intrusion detection. *arXiv preprint arXiv:1802.09089*.

Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. *Ieee Access*, *5*, 21954-21961.

Zaidi, S. K. A., Soomro, A. A., Ahmad, B., Hafeez, S., Majeed, M. K., Hussain, S. S., ... & Abbasi, M. D. Advanced AI-Driven Architecture for Real-Time Monitoring and Intelligent Fault Detection of Aircraft Engine Compressor and Fuel Systems under Emergency Operating Conditions.

Hodo, E., Bellekens, X., Hamilton, A., Dubouilh, P. L., Iorkyase, E., Tachtatzis, C., & Atkinson, R. (2016, May). Threat analysis of IoT networks using artificial neural network intrusion detection system. In *2016 International symposium on networks, computers and communications (ISNCC)* (pp. 1-6). IEEE.

Lippmann, R. P., & Cunningham, R. K. (2000). Improving intrusion detection performance using keyword selection and neural networks. *Computer networks*, *34*(4), 597-603.

Haq, R. U., Aman, F., Majeed, M. A., Raza, S., Khan, A., Hussain, R., & Majeed, M. K. (2025). DEVELOPING EDGE COMPUTING SOLUTIONS FOR IOT DEVICES TO REDUCE LATENCY AND ENHANCE REAL-TIME DECISION-MAKING. *Spectrum of Engineering Sciences*, 961-970.

Biggio, B., & Roli, F. (2018, October). Wild patterns: Ten years after the rise of adversarial machine learning. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security* (pp. 2154-2156).

Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., & Elovici, Y. (2018). N-baiot—network-based detection of iot botnet attacks using deep autoencoders. *IEEE Pervasive Computing*, *17*(3), 12-22.

Gul, L., Salim, I., Imran, M., Majeed, M. K., & Zia, A. An Intelligent CNN–LSTM Hybrid Deep Learning-Based Intrusion Detection Framework for Zero-Day Attack Detection in Network Traffic.

Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, *2*(1), 1-22.

Ring, M., Wunderlich, S., Scheuring, D., Landes, D., & Hotho, A. (2019). A survey of network-based intrusion detection data sets. *Computers & security*, *86*, 147-167.

Akbar, K., & Adeel, F. ARTIFICIAL INTELLIGENCE, DESIGN TIME AND RUN TIME METHODS FOR MOBILITY OF USERS INTERFACE.

Ali, S. F. A., Masroor, S., Shaikh, M. K., Jabeen, G., Naz, S. A., Aqeel, A., & Anjum, K. (2026). Biogenesis of Candida glabrata-Mediated Silver Nanoparticles: Characterization and Antibacterial Effectiveness Against Human Pathogenic Bacteria. *International Journal of Molecular Sciences*, *27*(3), 1263.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Akbar, K., Abrar, K., & Khan, S. A. (2022). Effect of Information and Communication Technologies (ICT) as Innovation Tool on Business Performance: Evidence from Pakistan. *Annals of Human and Social Sciences*, *3*(3), 494-504.

Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794).

Martins, J. N., Ensinas, P., Hovhannisyan, N., Chan, F., Babayeva, N., Berti, L., ... & Versiani, M. A. (2026). The Global Periapical Health Study (GPHS): a big data CBCT analysis of periapical pathology across 54 countries. *Journal of endodontics*.

Kate, A. (2026, April). *Integrating the NIST AI Risk Management Framework with Cybersecurity Framework Profiles for Sector-Specific Critical Infrastructure (Energy, Water, Transportation)*.

Raza, A., Ali, A. K. S., & Hussain, A. A. (2024). AI-driven approaches to cyber and information security: Machine learning algorithms for threat prediction and anomaly detection. *Spectrum of Engineering Sciences*, 565-573.

Khalaf, N. Z., Barazanchi, A., Ibraheem, I., Radhi, A. D., Shah, P., & Sekhar, R. (2025). Development of real-time threat detection systems with AI-driven cybersecurity in critical infrastructure. *Mesopotamian Journal of CyberSecurity*, *5*(2), 501-513.