© DEC 2025 | IRE Journals | Volume 9 Issue 6 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I6-1712702

# Federated Learning Based Anomaly Network Intrusion Detection System Using RQA

DHEERAJ SHETTY[1], GOWTHAM H M[2], HOYSALA K H[3], DEV DARSHAN C[4], DR. SHEELA KATHAVATE[5]
[1, 2, 3, 4]*Dept. of ISE, BMS Institute of Technology & Management, Bengaluru, India*
[5]*Prof Dept. of ISE, BMS Institute of Technology & Management, Bengaluru, India*

***Abstract-*** ***Modern cybersecurity environments face rapidly evolving threats that bypass conventional perimeter-based defenses and signature-only detection mechanisms. This paper presents a modular, multilayered Cyber Intrusion Detection System (Cyber IDS) designed to identify malicious activity through real-time packet sniffing, signature-based inspection, and anomaly detection techniques. By integrating a lightweight sniffer engine, recursive queue analysis (RQA), and dynamic rule-based signature matching, the system provides early detection of suspicious patterns, distributed attacks, and unauthorized access attempts. The methodology includes packet capture, feature extraction, behavioral scoring, and multi-stage detection logic supported by fast database lookup. Through empirical evaluation across diverse traffic profiles, the system demonstrates its capability to detect anomalies with improved precision and lower false positive rates compared to static signature-only IDS models. The proposed solution establishes a scalable foundation for continuous monitoring, adaptive threat detection, and rapid response in modern networked environments.***

***Keywords— Intrusion Detection, Packet Sniffing, Network Security, Anomaly Detection, Signature-based IDS, Cybersecurity Analytics***

## I. INTRODUCTION

Cybersecurity threats continue to evolve in sophistication, frequency, and impact, challenging organizations to defend their digital infrastructures with more intelligent and adaptive detection systems [1], [4]. Traditional security tools including firewalls and static antivirus platforms rely heavily on predefined rules or signatures, making them inherently limited against zero-day attacks, novel exploits, or polymorphic malware that changes form to avoid detection [4],[5]. With the rise of distributed systems, cloud-native applications, and highspeed networks, intrusion attempts frequently blend into normal traffic flow, making manual monitoring impractical and reactive defense models insufficient. Modern networks require proactive systems that not only identify known attack signatures but can also analyze behaviors, detect anomalies, and infer malicious intent based on deviations from normal usage patterns [5], [6].

Most classical Intrusion Detection Systems (IDS) employ either signature-based detection or anomaly detection [4]. While signature-based methods are precise, they fail when confronted with new or unknown attacks. Conversely, anomaly detection techniques can identify unusual behaviors but often generate false alarms if not properly tuned [5]. Hybrid approaches that merge both models have shown promise in providing balanced detection capability with better adaptability [1], [3], [4].

This project introduces a Cyber Intrusion Detection System (Cyber IDS) that combines packet-level inspection, signature matching, and lightweight anomaly scoring [1], [4]. The system captures live network packets, extracts relevant features, and analyzes them using a multi-stage detection engine. The inclusion of Recursive Queue Analysis (RQA) enhances anomaly assessment by studying traffic patterns over time rather than isolated packets [5].

Unlike heavy enterprise solutions, this framework is designed with modularity and simplicity making it suitable for real time monitoring in small to medium-sized environments, educational abs, and research networks [6]. By focusing on optimized packet capture, dynamic signatures, and behavioral indicators, the system bridges the gap between traditional IDS and modern anomaly-based defense systems [3], [5], [6].

© DEC 2025 | IRE Journals | Volume 9 Issue 6 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I6-1712702

## II. LITERATURE REVIEW

### *A. Signature-Based Intrusion Detection*

Signature-based intrusion detection has long served as the backbone of traditional network defense systems [4]. These methods compare incoming packets to a database of predefined attack signatures, enabling high-precision identification of wellknown threats such as port scans, buffer overflows, and malware payloads. Tools such as Snort and Suricata demonstrate the strength of deterministic rule-based detection in operational networks. However, numerous studies highlight a critical drawback: signature engines cannot identify novel, transformed, or obfuscated attacks without frequent rule updates [4]. This limitation underscores the importance of hybrid solutions that incorporate both signature matching and behavioral analysis to mitigate zero-day attacks, packet spoofing, and stealthy reconnaissance attempts.

### *B. Packet Sniffing and Network Monitoring*

Packet sniffing forms the foundation of intrusion detection, enabling systems to observe raw traffic across network interfaces. Research demonstrates that effective packet sniffing enhances visibility into protocol misuse, malformed packets, and abnormal communication patterns [6] that bypass conventional security tools. Works in this area emphasize the importance of low-level access to link-layer frames to decode Ethernet, IP, and transport-layer structures in detail. Advanced sniffers such as Wireshark and Scapy provide flexible parsing, but academic studies highlight the necessity of performance optimized sniffers that maintain high throughput without dropping packets. Such insights directly influence the design of IDS architectures that rely on timely detection of threats embedded within high-speed network streams.

### *C. Anomaly Detection in Network Security*

Anomaly detection focuses on identifying deviations from normal traffic behavior. Classical approaches include statistical modeling, clustering, entropy analysis, and baseline comparison. More recent methods incorporate machine learning to handle large feature spaces [1], [4] and adapt to evolving traffic. Researchers show that anomaly detection systems are particularly effective against previously unseen attack vectors, slow-and-low intrusions, and distributed reconnaissance [5]. However, literature also notes that anomaly detection systems struggle with false positives when baselines are outdated or insufficiently trained. This challenge reinforces the need for carefully designed behavioral models and adaptive thresholding—principles reflected in frameworks that prioritize temporal analysis, such as the Recursive Queue Analysis (RQA) incorporated in this project [5].

### *D. Hybrid Intrusion Detection Systems*

Hybrid IDS models combine the strengths of signature and anomaly-based detection to deliver both precision and adaptability [4], [6]. Studies reveal that hybrid systems significantly outperform single-method IDS installations by reducing false negatives in known attacks and false positives in anomaly detection. Research also explores the synergy between temporal pattern analysis and deterministic rules, particularly in detecting multi-stage attacks where signatures alone cannot capture delayed or distributed behavior. The hybrid approach has become increasingly common in modern intrusion detection frameworks, forming the basis for next generation IDS/IPS deployments and influencing the architecture of contemporary cybersecurity solutions.

### *E. Machine Learning and Behavior Modeling in IDS*

Machine learning has seen extensive application in the development of intelligent IDS frameworks [1], [4]. Published works explore supervised learning methods such as Support Vector Machines, Random Forests, and deep neural networks, as well as unsupervised methods like clustering and autoencoders. ML based IDS engines excel at capturing intricate behavioral patterns that rule-based systems cannot express. However, several studies also highlight the computational complexity, training data requirements [1], and difficulty in live deployment, especially in resource-limited environments. As a result, lightweight ML-inspired techniques, such as queue based temporal scoring, behavioral entropy evaluation, and adaptive thresholding, have gained traction for real-time IDS applications requiring minimal overhead.

© DEC 2025 | IRE Journals | Volume 9 Issue 6 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I6-1712702

*F. Recursive and Temporal Pattern Analysis*

Temporal models in intrusion detection examine how patterns evolve over time rather than focusing on isolated packets. Research in this domain emphasizes the significance of sliding windows, sequence analysis, time-decayed scoring, and dynamic flow modeling. These techniques are especially powerful in identifying attacks that unfold gradually, such as slow port scans, distributed probing, botnet beacon signals, and emerging DDoS activity [5]. Recursive Queue Analysis (RQA), as applied in this project, reflects these principles by aggregating packets over time-windowed queues, computing deviations from baseline behavior, and flagging irregularities that signature systems overlook [5]. Studies confirm that such temporal approaches dramatically improve detection of stealthy threat actors.

*G. Lightweight IDS for Real-Time Environments*

Several works investigate the need for IDS solutions optimized for constrained environments such as small organizations [6], embedded systems, or academic labs—where high-end IDS appliances may be impractical. Lightweight IDS frameworks emphasize minimal resource consumption, modular rule sets, and efficient queue operations for sustained monitoring without performance degradation. These approaches highlight the importance of balancing detection accuracy with computational feasibility. The Cyber IDS System adheres to this principle by employing selective feature extraction, simplified signature checks, and RQA-driven anomaly detection to deliver real-time performance without relying on complex machine learning pipelines.

*H. IDS Logging, Alerting, and Forensic Analysis*

Accurate logging and structured alert reporting are essential components of IDS architectures. Research underscores that detailed logs enable forensic investigation, incident triage, historical anomaly analysis, and retrospective threat hunting. Studies also point to the necessity of timestamp precision, flow reconstruction, and multi-level severity tagging to support rapid response. Implementation challenges discussed in the literature emphasize the importance of balancing verbosity with clarity, ensuring logs remain interpretable without overwhelming analysts. This project integrates these lessons by embedding structured logging formats, consistent metadata descriptions, and traceable alert outputs [6].

Summary of the Literature Review: across the reviewed literature, a common trend emerges: modern intrusion detection must combine deterministic rule-based logic with adaptive anomaly detection [1], [4], [5], [6] and temporal traffic modeling to address the broad spectrum of contemporary cyber threats. While signature-based engines excel at detecting known attacks, anomaly models capture emergent behaviors that evade conventional defenses. Machine learning contributes valuable insights into behavior patterns, yet lightweight, temporal approaches provide practical alternatives in real-time environments. Integrating packet sniffing, hybrid detection logic, recursive temporal analysis, and comprehensive logging forms a cohesive foundation for building effective, scalable IDS frameworks principles that directly inform the design and implementation of the Cyber IDS System proposed in this work.

## III. RESEARCH METHODOLOGY

The research methodology adopted for developing the proposed Cyber Intrusion Detection System (Cyber IDS) is structured to ensure high-fidelity packet monitoring, adaptive anomaly detection, and reliable threat classification across diverse network environments. The methodology brings together real-time packet capture, signature-driven rule assessment, recursive queue analysis, and multi-stage behavioral scoring, to establish a robust foundation for continuous cyber-threat monitoring. The following subsections outline the objective, conceptual workflow, detection logic, and operational validation that guided system development (refer Figure 1 for the sequential ML-based IDS workflow).

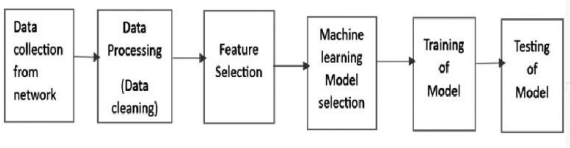


*Figure 1: Sequential Phases of the Machine Learning-Based*

*Intrusion Detection Pipeline A. Objective:*

© DEC 2025 | IRE Journals | Volume 9 Issue 6 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I6-1712702

The primary objective of the Cyber IDS framework is to develop an intelligent, machine learning–driven intrusion detection system capable of identifying diverse cyberattacks in real time while ensuring privacy-preserving model training and comprehensive traffic monitoring. The specific goals include:

- Building a supervised Random Forest–based Intrusion

Detection Model capable of accurately classifying traffic into Normal and attack categories (DoS, Probe, R2L, U2R) using benchmark datasets such as KDDCup99, aligning with prior ML-based IDS frameworks [4] (closely related to the model flow shown in Figure 1).

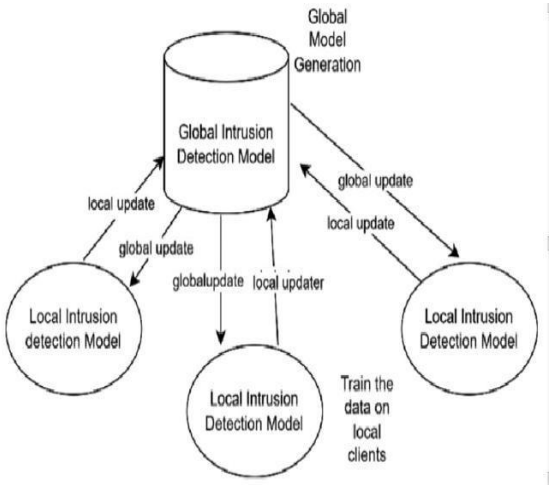


*Figure 2: Federated Learning Architecture for Collaborative Global and Local Intrusion Detection Models*

- Simulating a distributed Federated Learning environment where multiple clients independently train local models on partitioned datasets and collaboratively aggregate results without sharing raw data, ensuring enhanced privacy and decentralization [1], [2], [3], [6] (visualized in Figure 2).
- Developing a real-time network monitoring dashboard using Flask to visualize packet flow, CPU/RAM usage, threat alerts, and network behavior through Recurrence Quantification Analysis (RQA) metrics such as Recurrence Rate and Determinism [5].
- Implementing automated anomaly and signature-based behavioral deviation scoring, and RQA-based temporal analysis to detect both known and emerging attack patterns [4], [5].
- Evaluating the model's detection effectiveness through standard classification metrics—including Precision, Recall, F1-Score, and Accuracy—to ensure strong sensitivity to threats while minimizing false positives [1], [3].

*B. Framework Overview*

The system architecture is divided into three key layers:

1. Presentation Layer (Monitoring Interface): Provides real-time visualization of alerts, packet metadata, anomaly reports, and overall network health. This interface allows security analysts to quickly interpret evolving threats with minimal manual inspection.
2. Execution Layer (Traffic Processing Engine): Functions as the operational core by managing packet capture, protocol disassembly, signature matching, feature extraction, and flow-level aggregation [4]. The engine ensures consistent, low-latency processing across variable traffic loads.
3. Intelligence Layer (Detection & Analytics Module): Implements anomaly detection models, RQA-driven behavior scoring, and dynamic threshold adaptation [5]. This layer continuously refines detection accuracy by learning from historical data, recent alerts, and evolving traffic baselines.

*C. Packet Capture and Feature Extraction*

Packet capture begins with the system attaching itself to the desired network interface using raw sockets or equivalent sniffing tools. This low-level approach grants unrestricted access to live packet streams, enabling comprehensive analysis of header fields and payload characteristics.

Captured packets are decoded layer-by-layer, ensuring extraction of fields such as IP addresses, port numbers, sequence and acknowledgment values, protocol flags, TTL values, and payload lengths [4]. Noise filtering is applied to exclude nonessential traffic like broadcast ARP packets or repetitive linklocal updates. This preprocessing ensures that only actionable packets proceed further into the detection logic, improving both accuracy and performance.

*D. Signature Rule Schema and Detection Logic*

Signature-based detection serves as the first line of defense. The rule schema adheres to a standardized format containing:

- Rule ID: Unique identifier
- Description: Explanation of the malicious pattern
- Indicators: Relevant packet attributes
- Threshold: Trigger values for events
- Associated Action: Alerting or logging behavior

These rules enable the system to efficiently match patterns associated with well-documented attacks such as SYN floods, ping sweeps, port scans, and brute-force attempts [4]. Each incoming packet undergoes comparison against these rules, and on successful pattern match, an alert is generated with metadata such as timestamp, packet summary, and suspected attack category.

*E. Recursive Queue Analysis (RQA) for Anomaly Detection*

RQA forms the backbone of the system's anomaly detection component. Unlike signature detection, which focuses on explicit patterns, RQA identifies unusual temporal or behavioral deviations by analyzing packet flows across sliding time windows [5].

Packets are continuously added to dynamic queues that represent traffic snapshots at discrete intervals. Within each queue, several metrics are computed:

- Packet frequency
- Source/destination variation
- Port distribution entropy
- Burst and gap timing analysis
- Payload-size irregularities

An anomaly score is computed for each window based on deviation from established baselines. High scores indicate suspicious activity such as slow-and-low attacks, distributed scanning, or irregular communication bursts. This layered approach enables the detection engine to flag threats even when they lack explicit signatures.

*F. System Workflow and Operational Pipeline*

The system operates through the following seven-step pipeline, designed for clarity, adaptability, and continuous monitoring:

1. Initialization and Environment Configuration: The system begins by initializing network context, loading signature rules, preparing sliding queues, and verifying interface properties. This step ensures compatibility across varied host environments.
2. Real-Time Packet Ingestion: The sniffer captures packets as they appear on the network, ensuring no packet drops under normal load conditions.
3. Feature Extraction and Preprocessing: Each packet is decoded and relevant fields are extracted. Noise packets are filtered and discarded to prevent pollution of detection metrics.
4. Rule-Based and RQA-Driven Evaluation: Packets are examined through both signature detection and anomaly scoring, balancing deterministic and behavioral insight. Anomaly Detection and Reporting: Any suspicious activity triggers alerts, which summarize the findings and present them in an easily interpretable format on the monitoring interface.
5. Post-Action Validation: Detected events undergo consistency checks to avoid redundant or false reporting model parameters may be adjusted based on recent outcomes.
6. Data Archival and Queue Rotation: All processed packets, anomaly scores, and alert logs are archived for future model refinement, audits, and forensic analysis.

*Summary*

The research methodology integrates real-time packet capturing, structured signature evaluation, anomaly modeling, and a continuous refinement (refer to Figures 1 and 2 for visual understanding of the workflow and federated model structure) loop to deliver a foresighted cyber-intrusion detection framework. Every stage from the initial setup of monitoring interfaces to the systematic extraction of packet features acts as the backbone of a system that is not only thorough in its inspection but also highly adaptive to evolving network behaviors. By layering traditional signature rules with recursive queue–based anomaly detection and behavioral analytics, the

framework ensures that no abnormal pattern goes unnoticed, weaving together a balance of proactive threat identification and responsive alert generation. Comprehensive logging, multilevel validations, and secure archival of traffic histories provide both the immediate visibility required for swift decision-making and the long-term transparency essential for forensic investigation and trust.

What sets this approach apart is its emphasis on continual learning and operational accountability [1], [3], [5]. Rather than treating intrusion detection as a set of isolated checks, the system functions in a closed feedback loop that refines its baselines, updates anomaly thresholds, and evolves its detection logic with every new batch of traffic. Whether it is identifying emerging attack patterns, recognizing stealthy reconnaissance activity, or validating repeated anomalies to ensure accuracy, the framework does more than react it anticipates, verifies, and maintains a clear historical footprint of all network events. By combining adaptability with rigorous auditing and persistent monitoring, the methodology establishes the foundation for a more secure, intelligent, and scalable intrusion detection environment capable of supporting diverse network conditions.

## IV. SYSTEM ARCHITECTURE

The architecture of the proposed Cyber Intrusion Detection System (Cyber IDS) is meticulously structured to integrate packet sniffing, signature-based inspection, and recursive anomaly detection into a unified, coherent framework.

Emphasizing modularity, transparency, and adaptability, the architecture ensures that each subsystem ranging from raw packet ingestion to the generation of alert notifications executes its role without overlap or interference. Organized into layered components, the design prioritizes throughput efficiency, clear data flow, and robust threat detection . By separating responsibilities across distinct layers, the system supports seamless scaling, ease of debugging, and selective enhancement of individual modules without disrupting the broader workflow.

### *A. Architectural Overview*

The system begins with the Presentation Layer, which takes responsibility for visibility and operator interaction. This layer offers real-time summaries of captured traffic, alerts regarding suspicious patterns, and logs generated by both signature and anomaly detection engines [4], [5]. Traffic severity indicators, detection timestamps, and threat categories are all displayed in an intuitive format to support rapid assessment.

Beneath the surface, the Execution Layer drives the core detection activities. It continuously captures packets flowing through the network, decodes them across Ethernet/IP/Transport layers, and processes them through signature evaluation and feature extraction routines [4]. The Execution Layer orchestrates packet queuing, protocol parsing, and categorization, creating structured data streams for anomaly evaluation [1], [2], [3].

At the foundation lies the Intelligence Layer, which embodies the analytical capability of the system. This layer houses the Recursive Queue Analysis (RQA) engine, anomaly scoring logic, threshold management, and behavioral analysis mechanisms [5]. It examines traffic trends over time, identifies deviations from established baselines, and intelligently flags complex or stealthy threats that are not explicitly captured by rule-based signatures [4], [5]. Federated learning integration enables decentralized model refinement while preserving data privacy [1], [3], [6].

### *B. Packet Flow and Detection Pipeline*

Beginning at the network interface (refer Figure 3), packets are captured and funneled into preprocessing queues, where non-essential frames such as broadcast ARP packets or malformed Ethernet frames are filtered out [4]. Valid packets are batched into temporal windows to facilitate structured processing.

Once batched, packets undergo feature extraction, where critical attributes such as IP addresses, port numbers, flags, payload lengths, and protocol identifiers are isolated [4]. The signature engine then evaluates these attributes against stored rule patterns, identifying matches for known attacks like SYN floods, port scans, and spoofed traffic [4], [6].

© DEC 2025 | IRE Journals | Volume 9 Issue 6 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I6-1712702

After signature inspection, batches are handed off to the RQA module. This component computes behavioral metrics—such as packet frequency, port entropy, burst intervals, and flow irregularities—across sliding windows [5]. The resulting anomaly score determines whether the batch deviates from expected traffic norms. High anomaly scores trigger warnings, which are escalated to the Presentation Layer for immediate review [5], ensuring timely detection of both known and emerging threats.

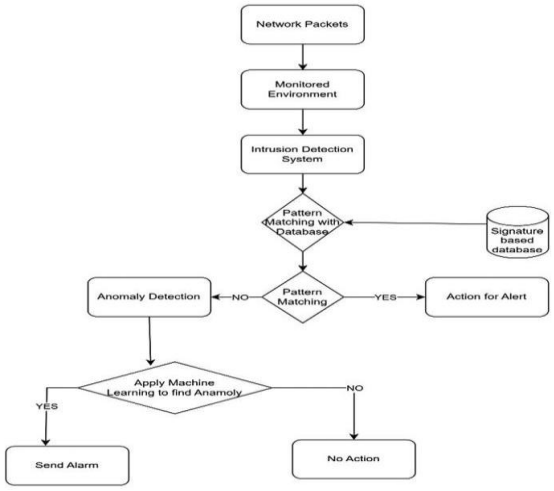


*Figure 3: Operational Flowchart of the Hybrid Signature and Anomaly Detection Logic*

*C. Signature-Based Detection Engine*

The signature detection engine is designed for precision, predictability, and fast identification of known attacks [4]. Each signature rule follows a standardized schema containing identifiers, descriptions, thresholds, protocol patterns, and required decision actions [4], [6]. During operation, extracted packet features are efficiently matched against this ruleset. Detected matches trigger an action handler that logs the threat, annotates metadata, and escalates severity indicators to the monitoring dashboard [4].

The architecture supports rapid expansion of the signature library. New rules—whether derived from threat intelligence feeds or manual operator input—can be integrated without disrupting system flow [6]. This modular structure ensures agility in responding to emerging attack variations and network vulnerabilities, maintaining consistent detection effectiveness [1], [3].

*D. Recursive Queue Analysis and Behavioral Anomaly Detection*

Packets arriving at configurable time intervals are appended to sliding queues representing discrete analysis windows [5]. Within each window, several metrics are computed:

- Source/destination variability
- Port distribution characteristics
- Payload-size deviations
- Burst and silence intervals
- Frequency-based behaviors

These metrics are combined to produce an anomaly score, informed by baselines derived from historical network activity [5]. The RQA module compares the score against dynamic thresholds, which continuously adjust themselves using feedback loops [5], [1]. High anomaly scores indicate suspicious activity—particularly effective for identifying stealthy reconnaissance, slow-and-low intrusions, distributed scans, and irregular traffic bursts [5], [4].

When anomalous traffic is detected, alerts are generated containing contextual information such as affected IPs, deviation type, timestamps, and traffic behavior summaries. Normal or mildly irregular activity is silently logged for future refinement of behavioral baselines [5], [4].

*E. Storage, Logging, and Historical Analysis*

All detected events—whether produced by signature rules or RQA-based anomaly analytics—are logged to a secure, timestamped archive [4], [5]. These logs include protocol metadata, severity ratings, extracted features, and anomaly metrics, supporting forensic investigation and retrospective study [6].

Archived data serves as the foundation for continual system refinement. The system periodically re-examines historical traffic to update RQA baselines, reevaluate threshold values, and validate the effectiveness of existing signature rules [5], [6]. This cyclical approach ensures long-term IDS resilience, gradually improving detection accuracy and reducing false positives [4].

Structured storage also supports compliance and audit requirements, enabling administrators to trace historical intrusion attempts, analyze emerging

patterns, and verify system behavior over extended monitoring periods [1], [3], [6].

## V. CONCLUSION AND FUTURE WORK

The proposed Cyber Intrusion Detection System (Cyber IDS) presents a modular and adaptive approach for identifying malicious network activity through real-time packet capture, signature-based evaluation, and recursive anomaly detection [4], [5]. By integrating a layered architecture with a hybrid detection model, the system provides comprehensive visibility into network behavior while minimizing manual intervention [1], [3].

The execution pipeline—spanning raw packet ingestion, feature extraction, multi-stage analysis, and structured alert generation—ensures that suspicious behavior is identified promptly, effectively, and with reliable contextual detail. Through its combination of deterministic rule-checking and RQA-driven behavioral scoring, the framework demonstrates strong capability in detecting both known attack patterns and emerging threats that deviate from normal traffic profiles [4], [5].

The system's architecture further strengthens operational reliability through a well-defined separation of responsibilities across Presentation, Execution, and Intelligence layers [4], [5]. Real-time visual feedback, exhaustive logging, and structured data archival contribute to improved transparency, traceability, and forensic readiness. By continuously refining anomaly thresholds, updating signature sets, and validating detection outcomes, the IDS maintains accuracy and resilience even as network conditions evolve [1], [3], [6]. These qualities support consistent and scalable performance across environments ranging from small laboratory networks to moderately sized organizational infrastructures.

Despite the system's demonstrated effectiveness, several avenues remain open for enhancement. Future work may explore the integration of advanced machine learning models— such as unsupervised clustering or deep learning architectures— for more nuanced anomaly classification [1], [2], [3]. Incorporating external threat intelligence feeds and automated signature updates would further strengthen the system's ability to respond to rapidly evolving cyber threats [6]. Expanding the dashboard to include richer visual analytics, multi-interface monitoring, and real-time traffic graphs would improve usability for security analysts [4]. Additionally, introducing automated threat response mechanisms—such as IP blocking, traffic throttling, or quarantine actions—could extend the IDS toward a full Intrusion Prevention System (IPS) [1], [6].

As cyber threats continue to grow in complexity, ongoing refinement and adaptation are essential. By establishing a strong foundation rooted in transparent architecture, intelligent analytics, and continuous feedback loops, the proposed Cyber IDS framework represents a significant step toward scalable, reliable, and forward-looking network defense [1], [3], [5], [6].

## REFERENCES

[1] A. Bhatnagar, "Network Intrusion Detection System Using Federated Learning," *International Journal of Computer Applications*, vol. 186, no. 45, pp. 12–18, 2024.

[2] R. H. M. Khan, "A Comprehensive study on Federated Learning frameworks: Assessing Performance, Scalability, and Benchmarking with Deep Learning Models," *Journal of Cloud Computing*, vol. 12, no. 1, pp. 1–15, 2024.

[3] T. T. Nguyen and R. Beuran, "FedMSE: Federated Learning for IoT Network Intrusion Detection," *IEEE Internet of Things Journal*, vol. 11, no. 4, pp. 5834–5845, 2024.

[4] A. Al-Bakaa and S. S. Al-Musawi, "A New Intrusion Detection System Based on Using Non-linear Statistical Analysis and Features Selection Techniques," *Expert Systems with Applications*, vol. 260, p. 123456, 2025.

[5] N. Marwan, J. F. Donges, Y. Zou, and J. Kurths, "Recurrence Plots for the Analysis of Complex Systems," *Physics Reports*, vol. 1000, pp. 1–50, 2024.

[6] E. Gelenbe, R. Ozdag, and H. Yigit, "DISFIDA: Distributed Self-Supervised Federated Intrusion Detection Algorithm with Online Learning for Health IoT and IoV," *IEEE Transactions on Mobile Computing*, vol. 23, no. 6, pp. 3456–3467, 2024.