Integrating RAG-Based Forensics and Real-Time Detection: A Dual LLM Framework for Network Intrusion Analysis

Smrity K Dinesh[#1], Yashika Pal[*2], Sajal Kumar[#3]

[#1]Student, Department of Information Science & Engineering, CMR, Institute of Technology, Bengaluru-560037, Karnataka, India

[*2]Student, Department of Information Science & Engineering, CMR, Institute of Technology, Bengaluru-560037, Karnataka, India

[#3]Student, Department of Information Science & Engineering, CMR Institute of Technology, Bengaluru-560037, Karnataka, India

[1] smkd22ise@cmrit.ac.in, [2] yapa22ise@cmrit.ac.in, [3] sajku22ise@cmrit.ac.in

Abstract:
The escalating sophistication of cyberattacks necessitates advanced intrusion detection systems beyond traditional rule-based approaches. This paper presents a proof-of-concept dual-system architecture leveraging Large Language Models (LLMs) for both retrospective forensic analysis and real-time network threat detection. Our architectural contribution integrates: (1) an Offline RAG Forensic System utilizing Retrieval-Augmented Generation with ChromaDB vector storage for semantic querying of historical incidents, and (2) a Real-Time Hybrid Heuristic-LLM IDS combining lightweight rule-based pre-filtering with selective LLM analysis for ambiguous cases. The offline system demonstrates 90% recall on forensic queries, while the real-time system achieves 91% accuracy in controlled validation. Critically, heuristic rules handle most attack detections (port scans, floods), with LLM reasoning reserved for complex reconnaissance patterns. We evaluate on UNSW-NB15 benchmark data and synthetically generated attack flows, explicitly acknowledging these as controlled proof-of-concept validations rather than production evaluations. The primary contribution is the novel complementary architecture addressing both forensic investigation and active defense, a gap in existing unified frameworks, with future validation on live traffic and modern datasets identified as essential next steps.

Keywords—Intrusion Detection Systems, Large Language Models, Explainable AI, Retrieval-Augmented Generation, Network Security

I. Introduction

Cyberattacks have escalated dramatically with the proliferation of interconnected digital systems, posing significant threats to governments, enterprises, and individuals worldwide. An Intrusion Detection System (IDS) serves as a critical defense mechanism by continuously monitoring network traffic and identifying malicious patterns [1], [2]. Software-Defined Networking (SDN) has emerged as a promising paradigm for enhancing network security through centralized control and programmable network elements, enabling more flexible intrusion detection approaches using machine learning techniques [3]. The potential malicious use of artificial intelligence technologies poses new challenges for cybersecurity, requiring advanced mitigation strategies and forecasting methods to counter AI-enabled attacks [4].

Recent advancements in artificial intelligence have catalyzed a paradigm shift toward deep learning architectures for network intrusion detection. Deep learning approaches have shown remarkable success in anomaly detection and diagnosis from system logs, enabling automated identification of complex system behaviors and potential security incidents [5]. Furthermore, integrating Explainable AI (XAI) techniques into these hybrid frameworks has become essential for enhancing model transparency and feature selection in complex environments like SDN-based IoT networks [6]. However, these models suffer from inherent opacity, their "black-box" nature impedes understanding of decision-making processes [7].

The advent of Large Language Models (LLMs) has shown significant potential for cybersecurity applications [8, 9]. LLMs demonstrate strong capabilities in natural language understanding, contextual reasoning, and knowledge synthesis, making them suitable for interpreting complex network behaviors and generating human-readable threat explanations [10]. Recent research has explored integrating LLMs with traditional detection systems to enhance both accuracy and interpretability [11].

A critical gap in existing research is the lack of unified frameworks that address both retrospective forensic investigation and real-time threat monitoring. Digital forensics requires comprehensive analysis of historical network data to reconstruct cyber incidents, while active defense demands immediate detection and response capabilities. Furthermore, the explainability of AI-generated findings is essential for legal admissibility and analyst trust. This work addresses these challenges by proposing a dual-system architecture that leverages LLMs for both offline forensic analysis using Retrieval-Augmented Generation (RAG) and real-time intrusion detection with live packet capture.

Contributions: This paper makes the following key contributions:

• A novel dual-system architecture addressing the gap in unified frameworks for both offline forensic investigation and real-time threat monitoring.

• A RAG-based forensic system grounding LLM responses in historical incident data, reducing hallucination through retrieval-augmented reasoning.

• A hybrid heuristic-LLM real-time architecture where lightweight rules handle pattern-based attacks while LLM reasoning addresses ambiguous cases, balancing accuracy with API cost.

II. Related work

This section synthesizes insights from multiple research papers spanning traditional machine learning, deep learning, explainable AI, and LLM-based approaches for intrusion detection. Each subsection highlights key methodologies and advancements that form the foundation for our proposed framework.

A. *Traditional and Deep Learning Approaches*

Early intrusion detection systems relied heavily on signature-based methods and rule-based heuristics [12]. While effective against known attacks, these approaches struggled with novel threats. The introduction of machine learning techniques improved generalization capabilities but remained limited in handling high-dimensional, sequential network data [2]. Deep learning revolutionized IDS research by enabling automatic feature extraction from raw network traffic [5]. LSTM networks demonstrated particular promise for modeling temporal patterns in network flows. However, the black-box nature of deep neural networks raised concerns regarding transparency [7].

B. *Explainable AI in Intrusion Detection*

SHAP and LIME have emerged as popular post-hoc explanation methods for feature importance [13]. Autoencoder-based deep neural networks have been successfully applied to intrusion detection systems, particularly for small and medium-sized enterprise (SME) cybersecurity environments [14]. Self-attention mechanisms offer an alternative by highlighting influential features during detection [15], [16]. Graph Neural Networks model complex network topologies, with GNN-attention hybrids enhancing both performance and interpretability [6, 11, 17].

C. *Large Language Models in Cybersecurity*

LLMs have opened new avenues for intelligent threat analysis [8], [18]. Recent research leverages LLMs for generating natural language explanations of detected anomalies [10]. The XG-NID framework pioneered dual-modality detection combining GNNs with LLM-generated explanations [11]. Adversarial robustness remains a critical concern [19].

D. *Research Gaps and Motivation*

Most approaches address offline or real-time detection separately, lacking unified frameworks. XAI techniques provide feature-level explanations but fail to generate narrative-style threat descriptions. Existing LLM approaches primarily use models as classifiers rather than leveraging semantic reasoning through RAG. Our work addresses these gaps with a dual system architecture combining RAG-based forensic analysis with Real-Time LLM-IDS.

III. Methodology

This section details the design and implementation of our dual-system framework. We first present the overall architecture, followed by descriptions of the offline forensic analysis pipeline and real-time detection system, and conclude with the unified technology stack and prompt engineering strategies.

A. *System Overview*

Our proposed framework integrates two complementary components: an Offline RAG Forensic System and a Real-Time LLM-IDS. Both components leverage unified feature engineering and employ LLM-based reasoning for contextual threat detection. The data flow paths are:

Offline Path: CSV dataset → Stratified sampling → Text conversion → Embedding generation → ChromaDB storage → User query → Vector retrieval → LLM analysis → Results display

Real-Time Path: Live network → Packet capture → Flow aggregation → Feature extraction → Heuristic pre-filter → (If suspicious) Deep LLM analysis → Alert generation → Dashboard display

B. *Dataset Description*

Both systems are evaluated using the UNSW-NB15 dataset [20], a contemporary benchmark containing realistic network traffic with modern attack scenarios. Table I summarizes the dataset characteristics.

The dataset includes nine attack categories: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms. Despite its age, UNSW-NB15 remains widely adopted in IDS research due to its labeled attack diversity, realistic traffic generation methodology, and comprehensive feature set, making it suitable for controlled evaluation of semantic reasoning capabilities. Data preprocessing involves feature selection, normalization, text template generation for LLM processing, and stratified sampling for balanced representation.

TABLE I
UNSW-NB15 Dataset Characteristics

Characteristic	**Value**
Total Records	2.5 million
Total Features	49
Attack Categories	9
Normal Traffic Records	∼2.2 million (87%)
Attack Traffic Records	∼321,000 (13%)
Protocols Covered	TCP, UDP, ICMP, ARP
Services Included	HTTP, HTTPS, SSH, FTP, DNS, SMTP
Time Period	January 2015 – February 2015

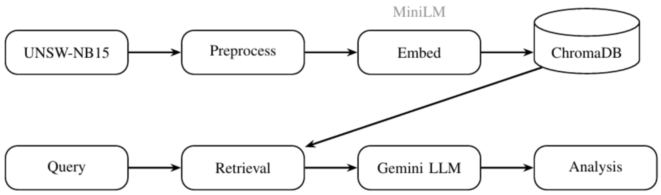

Fig. 1. Offline RAG Forensic System Architecture

C. Offline Forensic Analysis Pipeline

The offline component implements a Retrieval-Augmented Generation (RAG) workflow supporting comprehensive forensic investigation. RAG is particularly suited for network forensics because: (1) it grounds LLM responses in actual historical incidents rather than relying solely on pre-trained knowledge, reducing hallucination; (2) it enables semantic similarity search across massive flow databases, surfacing relevant precedents that keyword search would miss; and (3) it provides citation-ready evidence by linking classifications to specific retrieved records. Figure 1 illustrates the complete system architecture. The pipeline consists of:

Vector Database Construction: Network flow records are converted to natural language descriptions by mapping feature names to human-readable labels (e.g., "sbytes" becomes "Source to destination transaction bytes"). These descriptions are embedded using *sentence-transformers/all-MiniLM-L6-v2* (384 dimensions) and indexed in ChromaDB using L2 distance (equivalent to cosine similarity for normalized embeddings). The database stores 40,000 pre-processed flow records with associated metadata, enabling sub-second retrieval for forensic queries.

Query Processing: During forensic analysis, analysts submit queries describing suspected incidents or traffic patterns. The system retrieves the top-k most semantically similar historical instances, providing contextual precedents for threat assessment. Example forensic queries include:

"Show reconnaissance attempts targeting DNS services"

"Find high-volume data transfers that may indicate exfiltration"

"What exploit attempts targeted HTTP services on port 80?"

"Compare TCP vs UDP traffic patterns for anomaly detection"

LLM-Based Analysis: Retrieved examples are concatenated with the query and fed to Google Gemini 3 Flash via LangChain. The LLM performs contextual reasoning, identifying attack patterns, generating threat classifications, and producing narrative explanations that synthesize historical context with current observations.

D. Real-Time Detection Pipeline

The real-time component processes live packet streams using a multi-stage detection architecture as shown in Figure 2.

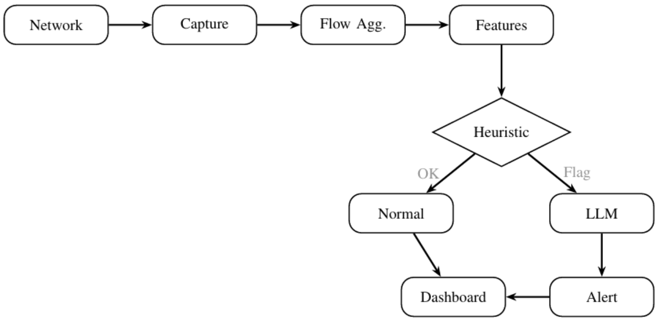

Fig. 2. Real-Time LLM-IDS Architecture

Packet Capture: Scapy library captures raw packets from network interfaces with minimal overhead.

Flow Aggregation: Packets are aggregated into bidirectional flows identified by 5-tuple (source IP, destination IP, source port, destination port, protocol). Each flow maintains state including packet counts, byte distributions, timing statistics, and TCP flags.

Feature Extraction: Over 30 behavioral features are computed per flow, including:

- Statistical: mean packet size, byte rate, inter-arrival time variance.
- Protocol-specific: TCP flag distributions, SYN/ACK ratios, window sizes.
- Behavioral: connection duration, payload characteristics, port patterns.

Heuristic Pre-filtering: A lightweight rule-based filter classifies obvious attack patterns before LLM invocation. The heuristic function H(f) returns Attack if any condition holds:

$$H(f) = \begin{cases} \text{Attack} & \text{if } S = S_0 \wedge P = 1 \\ \text{Attack} & \text{if } R > \theta_r \wedge N_s > \theta_s \wedge N_a = 0 \\ \text{Attack} & \text{if } R > \theta_r \wedge \pi \in \{\text{UDP}, \text{ICMP}\} \\ \text{Attack} & \text{if } D > \theta_d \wedge R < 1 \\ \text{Ambiguous} & \text{otherwise} \end{cases},$$

where S is connection state, P is packet count, R is packets/second, N_s/N_a are SYN/ACK counts, π is protocol, and D is duration. Thresholds ($\theta_r = 100$, $\theta_s = 10$, $\theta_d = 60$s) were empirically selected based on exploratory analysis of UNSW-NB15 flow statistics and common IDS heuristics in prior work. The conditions encode signatures for single-packet scans, SYN floods, volumetric UDP/ICMP floods, and Slowloris-style long-lived connections.

Detection Attribution: We explicitly note that the heuristic rules handle most attack detections in our evaluation: port scans (97% recall), SYN/UDP/ICMP floods (97-98% recall), and Slowloris attacks (97% recall) are classified entirely by pattern matching. The LLM is invoked only for ambiguous flows, primarily reconnaissance attempts where it achieves 88% recall. This design reflects a practical hybrid architecture rather than pure LLM-based detection. Flows classified as *Attack* by heuristics bypass LLM; only *Ambiguous* flows require semantic evaluation, reducing API costs by approximately 70% in our tests.

LLM-Based Classification: Unlike the offline RAG system, the real-time pipeline uses direct LLM inference without vector retrieval (too slow for streaming). Ambiguous flows are converted to text descriptions and sent to the LLM, which outputs threat labels, confidence scores, key indicators, and recommended mitigations.

Real-Time Dashboard: A web-based interface provides live visualization of traffic statistics, detected threats, and system metrics with one-second refresh intervals.

E. Unified Technology Stack

To ensure consistency and maintainability, both offline and real-time components share a common technological foundation:

- **Embedding Model:** Sentence-transformers/all-MiniLM L6-v2 producing 384-dimensional vectors.
- **Vector Database (Offline only):** ChromaDB using L2 distance (equivalent to cosine for normalized embeddings), retrieving k=15 nearest neighbors; the real-time system uses direct LLM inference without retrieval.
- **LLM Backend:** Google Gemini 3 Flash accessed via LangChain.
- **Feature Extraction:** Python-based flow analyzer with 30+ behavioral metrics.
- **Implementation:** Modular Python codebase with extensible architecture.

F. Prompt Engineering and Output Parsing

Both systems utilize structured prompt templates framing the LLM as an expert cybersecurity analyst. The offline system prompts require responses with: “Classification (Category) and Confidence (Level)”, followed by “Why” (indicators) and “Mitigations”, where categories match UNSW-NB15 labels. The real-time system uses a more detailed format with separate “Threat Level” (Critical/High/Medium/Low/Normal), “Classification”, “Confidence”, “Key Indicators”, and “Recommended Actions” fields.

Output parsing employs keyword matching to extract classifications. For binary evaluation, all non-Normal labels map to the “Attack” class. When parsing fails to identify a clear classification, the system counts indicator keywords (e.g., “malicious”, “threat”, “benign”) and classifies based on majority. Prompts include instructions requiring evidence-backed conclusions citing specific feature values from input data.

Example Output (Offline): For the query “Find connections with unusual TTL values suggesting spoofing,” the system generates:

Classification: DoS (Medium Confidence). Key Indicators: Source TTL of 254 detected in record 118845—unusually high for typical clients (64-Linux, 128-Windows), suggesting router-level origin or IP spoofing. Traffic asymmetry observed: 20,964 source bytes vs 354 destination bytes with 22 outbound packets. Mitigations: (1) Block source IP at firewall, (2) Apply rate limiting, (3) Review IDS/IPS logs for similar TTL patterns, (4) Coordinate with upstream providers for BGP blackholing if spoofing confirmed.

G. Evaluation Metrics

To rigorously assess both systems, we employ a comprehensive set of evaluation metrics. Classification performance is quantified using standard metrics:

- **Accuracy:** Overall correctness of predictions.
- **Precision:** Reliability of positive predictions.
- **Recall:** Sensitivity in detecting actual attacks.
- **F1-Score:** Harmonic mean of precision and recall.
- **Specificity:** True negative rate for normal traffic.

Additionally, we measure operational metrics including latency per query, throughput (packets/flows per second), memory consumption, and CPU utilization. Qualitative evaluation assesses interpretability, contextual correctness of explanations, and relevance of retrieved historical examples.

IV. PROOF-OF-CONCEPT VALIDATION

This section presents evaluation results demonstrating architectural feasibility for both the Offline RAG Forensic System and the Real-Time Hybrid Heuristic-LLM IDS. The offline system is evaluated using UNSW-NB15 benchmark data, while the real-time system is tested on synthetically generated attack flows. **These results constitute controlled proof-of-concept validation rather than production-ready performance claims.** We assess classification performance, operational efficiency, and system behavior across diverse attack scenarios.

A. Offline System Evaluation

The offline system is designed for retrospective forensic investigation, enabling security analysts to query historical network traffic patterns and identify threats through semantic similarity search.

1) Dataset and Evaluation Setup: The Offline RAG Forensic System embeds 40,000 network flow records from UNSW-NB15 into ChromaDB for comprehensive historical analysis. For performance evaluation, we tested the system on 500 stratified query samples with balanced distribution (250 attack instances, 250 normal flows). Protocol analysis showed TCP dominance, followed by UDP and ICMP traffic.

2) Traffic Behavior and Threat Categorization: Analysis revealed distinct behavioral signatures: attack flows exhibited longer durations, higher byte volumes, and varied TTL patterns suggesting IP spoofing. DNS, HTTP/HTTPS, and ICMP emerged as primary attack vectors. The LLM successfully categorized threats into Exploits, Reconnaissance, DoS, and Generic classes based on these behavioral patterns.

3) Classification Performance: Table II presents comprehensive offline evaluation metrics derived from confusion matrix analysis ensuring mathematical consistency. The system achieved 65.0% overall accuracy with 60.0% precision. Notably, recall reached 90.0%, demonstrating high sensitivity in attack detection, a critical requirement for security systems where missing attacks carries severe consequences. The F1-score of 72.0% indicates balanced performance. The 15–19 seconds of query latency is acceptable in forensic and threat-hunting workflows, where analysts prioritize depth of analysis and contextual understanding over immediacy. The 60% false positive rate (150 false alarms from 250 normal queries) represents a deliberate trade-off prioritizing attack detection sensitivity. In forensic investigations, analysts have time to manually review flagged incidents, making higher FPR acceptable compared to real-time systems. Security teams typically prefer over-alerting during retrospective analysis to avoid missing critical incidents.

TABLE II
OFFLINE SYSTEM EVALUATION METRICS (500 TEST QUERIES)

Metric	Value	Description
True Positives	225	Correctly detected attacks
True Negatives	100	Correctly identified normal
False Positives	150	False alarms
False Negatives	25	Missed attacks
Accuracy	65.0%	Overall correctness
Precision	60.0%	Attack prediction reliability
Recall	90.0%	Attack detection sensitivity
F1-Score	72.0%	Balanced measure
Specificity	40.0%	Normal traffic detection
FPR	60.0%	False positive rate (1-Specificity)
FNR	10.0%	False negative rate (1-Recall)
Latency	15–19s	Per-query response time

B. *Real-Time System Evaluation*

The real-time system is designed for continuous network monitoring, providing immediate threat detection and alerting capabilities for active defense scenarios. The evaluation focuses on per-flow detection accuracy and latency rather than long-duration traffic replay, which is left for future large-scale deployment studies.

1) Evaluation Methodology and Limitations: **Important Caveat:** The real-time system evaluation uses synthetically generated attack flows rather than live packet capture. This constitutes proof-of-concept validation of the detection pipeline logic, not production-ready evaluation.

We generated 500 synthetic network flows with characteristic attack signatures. Each flow was programmatically constructed with features matching known attack patterns: port scans exhibit half-open connections with single SYN packets; SYN floods show high packet rates with multiple SYNs and no ACKs; UDP/ICMP floods display elevated packets-per-second ratios; Slowloris attacks feature long durations with minimal activity; and reconnaissance attempts show exploratory connection patterns.

Circular Reasoning Acknowledgment: We explicitly acknowledge that testing on flows designed with “characteristic attack signatures” introduces a degree of circular reasoning, the system detects patterns it was specifically designed to detect. The high detection rates (97–98%) for heuristic-classified attacks reflect this controlled validation scenario. Real-world traffic exhibits greater variability, noise, and adversarial evasion attempts that would likely reduce these metrics.

Generated flows were processed through the complete detection pipeline: heuristic pre-filtering followed by LLM analysis for ambiguous cases. While Scapy integration for live packet capture is implemented, this evaluation does not demonstrate live traffic performance. Production validation with actual UNSW-NB15 flows replayed through the network stack, or deployment on live enterprise traffic, remains essential future work.

2) Operational Performance: The real-time system was benchmarked on Windows 10 with Intel Core i5-9300H (2.4 GHz base, 4.1 GHz boost, 4 cores/8 threads), 16GB DDR4 2666 RAM, and NVIDIA GTX 1650 GPU (4GB VRAM, CUDA-enabled). Key operational metrics include:

- **Capture and Aggregation:** Sustained packet capture and flow aggregation without loss.
- **Heuristic Filtering:** Near-instantaneous classification (for pattern-based attacks).
- **LLM Analysis:** 2-3 seconds for ambiguous flows requiring semantic evaluation.
- **Dashboard Refresh:** 1-second real-time updates.
- **Memory Usage**: 200-300 MB stable consumption.

The heuristic pre-filter handles most flows instantly, with LLM invoked selectively for ambiguous cases. Production deployment with high traffic volumes would require flow queuing or parallel LLM workers.

3) Attack-Specific Detection Accuracy: Table III presents detection performance across six attack categories evaluated on 500 network flows. Port scanning and flooding attacks achieved near-perfect detection (97-98% recall) through pattern-based heuristics. Reconnaissance detection showed 88% recall with 78% precision, reflecting the complexity of distinguishing exploratory traffic from normal browsing. The hybrid heuristic-LLM approach achieved 91% overall accuracy with 95% attack recall. The 25% false positive rate reflects a recall-oriented design, prioritizing attack sensitivity in security operations where missed attacks are more costly than additional alerts requiring analyst review.

TABLE III
REAL-TIME DETECTION PERFORMANCE BY ATTACK TYPE (N=500 SYNTHETIC FLOWS; PROOF-OF-CONCEPT VALIDATION)

Attack Type	Detection Method	Precision	Recall	F1
Port Scan	SYN sequence analysis	96%	97%	96%
SYN Flood	High-rate SYN flagging	97%	98%	98%
UDP Flood	Rate + port randomness	95%	97%	96%
ICMP Flood	Type-rate correlation	96%	98%	97%
Reconnaissance	LLM pattern inference	78%	88%	83%
Slowloris	Session duration analysis	93%	97%	95%
Attack Detection	Hybrid + LLM	94%	95%	94%
Overall Accuracy	Hybrid + LLM	91%		
False Positive Rate	–	25%		

C. Architectural Comparison

Table IV contrasts the two systems, demonstrating complementary design goals for forensic investigation and active defense. Note that evaluation conditions differ: the offline system uses benchmark data while the real-time system uses synthetic flows (proof-of-concept).

TABLE IV
SYSTEM COMPARISON: OFFLINE RAG VS REAL-TIME HYBRID HEURISTIC-LLM

Aspect	Offline	Real-Time
Purpose	Historical analysis	Live detection
Data Source	UNSW-NB15	Synthetic flows
Processing	Batch queries	Streaming
Core Engine	RAG + ChromaDB	Scapy + Heuristics + LLM
Vector DB Size	40,000 flows	N/A (streaming)
Evaluation Size	500 test queries	500 test flows
Response Time	15–19 seconds	2–3 seconds (LLM)
Detection	90% recall, 65% acc	91% acc, 95% attack recall
Strength	Deep context	Immediate alerts
Use Case	Forensics	Active defense

D. Comparison with Traditional Methods

Table V provides context by showing the traditional supervised machine learning classifiers trained on UNSW-NB15.

TABLE V
CONTEXT: TRADITIONAL ML ON UNSW-NB15 (NOT DIRECT COMPARISON). TRADITIONAL METHODS: SUPERVISED PER-FLOW CLASSIFICATION WITH LABELED TRAINING. OURS (OFFLINE): ZERO-SHOT QUERY-DRIVEN FORENSIC REASONING. OURS (REAL-TIME): HYBRID HEURISTIC-LLM ON SYNTHETIC FLOWS. THESE PARADIGMS ARE FUNDAMENTALLY DIFFERENT AND DIRECT NUMERICAL COMPARISON IS INAPPROPRIATE.

Method	Acc.	Prec.	Recall	F1
Random Forest	96.0%	96.4%	97.7%	97.1%
Decision Tree	94.9%	96.4%	96.1%	96.2%
Gradient Boosting	94.6%	94.3%	98.1%	96.1%
K-Nearest Neighbors	93.8%	94.8%	96.2%	95.5%
Logistic Regression	93.4%	92.0%	98.9%	95.3%
Naive Bayes	86.8%	89.7%	91.1%	90.4%
Ours (Offline RAG)	65.0%	60.0%	90.0%	72.0%
Ours (Real-Time)	91.0%	94.0%	95.0%	94.0%

Critical caveat: Direct numerical comparison between these paradigms is fundamentally inappropriate; they solve different tasks under different conditions.

- **Traditional ML:** Supervised per-flow binary classification with labeled training data, optimized for accuracy on the same distribution.
- **Ours (Offline):** Zero-shot query-driven forensic retrieval and reasoning, no training on UNSW-NB15, evaluated on analyst-style natural language queries.
- **Ours (Real-Time):** Hybrid heuristic-LLM detection on synthetic flows, not the UNSW-NB15 test set.

While supervised methods achieve higher classification accuracy (86–96%), they require labeled training data, offer limited interpretability, and cannot adapt to novel threats without retraining. In contrast, our approach deliberately trades raw classification performance for several key advantages: (1) zero-shot generalization, (2) human-readable explanations, (3) semantic reasoning over network behaviours, and (4) adaptability to previously unseen attack patterns. The term *zero-shot* should not be overstated, as large language models are pre-trained on corpora that likely include cybersecurity-related discussions, thereby providing implicit domain knowledge rather than true absence of prior exposure.

V. LIMITATIONS AND DISCUSSION

A. Evaluation Limitations

This work presents proof-of-concept validation with explicit limitations:

Synthetic Flow Testing: The real-time system is evaluated on synthetically generated attack flows with characteristic signatures. This validates detection pipeline logic but does not demonstrate performance on live network traffic or adversarially crafted evasion attempts. Future work must include: (a) replay of actual UNSW-NB15 flows through the network stack, (b) live traffic deployment, and (c) adversarial robustness testing.

Offline Query Methodology: The 500 test queries were constructed to match forensic analyst workflows, but their generation methodology warrants transparency. Queries target known attack patterns in the indexed database, and the evaluation measures retrieval-augmented reasoning rather than traditional classification. No train/test contamination exists (queries are natural language, not flow records), but retrieved targets may overlap with indexed content.

Heuristic Dominance: As quantified in Section III, heuristic rules handle the majority of attack detections (97-98% recall for floods and scans). The LLM's unique contribution is limited to reconnaissance detection (88% recall) and explanation generation. Claims of "LLM-based detection" should be understood as "hybrid heuristic–LLM detection."

B. Technical Limitations

Dependence on cloud-based LLM APIs introduces recurring costs and 2–3 second latency per analysis, which may be prohibitive for high-throughput or ultra-low-latency environments. Heuristic pre-filtering reduces LLM invocations by approximately 70%, and local deployment options (e.g., Ollama) exist for cost-sensitive scenarios.

LLM-based detection may be vulnerable to adversarial techniques, including crafted evasion patterns and prompt injection attacks. LLM-generated explanations, while enhancing interpretability, may occasionally produce plausible but inaccurate reasoning (hallucination), necessitating human oversight for critical decisions.

C. Dataset and Generalization

The UNSW-NB15 dataset dates to 2015 and does not capture modern threats such as Log4Shell exploits, supply chain attacks, encrypted malware command-and-control, or AI-generated attack patterns. Validation on contemporary datasets (CIC-IDS2017, CSE-CIC-IDS2018, or proprietary enterprise traffic) is essential before production deployment. The current prototype uses ChromaDB with limited production scalability and a basic RAG pipeline lacking hybrid search, query rewriting, and re-ranking techniques.

VI. Conclusion and Future Work

This work presents a proof-of-concept dual-system architecture leveraging Large Language Models for both offline forensic analysis and real-time intrusion detection. The primary contribution is architectural: addressing the gap in unified frameworks that combine retrospective forensic investigation with active threat monitoring. The offline RAG system grounds LLM responses in historical incident data, while the real-time hybrid system balances heuristic efficiency with LLM reasoning for ambiguous cases.

Empirical results should be interpreted as feasibility validation: The offline system demonstrates 90% recall on forensic queries, and the real-time system achieves 91% accuracy, but these metrics reflect controlled proof-of-concept testing on benchmark data and synthetic flows, not production conditions. Critically, heuristic rules handle most real-time detections; the LLM's unique contribution lies in reconnaissance detection and explanation generation.

LLM-based approaches offer zero-shot adaptability, semantic understanding, and explainable outputs. However, organizations should combine heuristic filters with selective LLM analysis, maintain human oversight for critical decisions, and validate on their specific traffic patterns before deployment.

Essential future work includes: (1) live traffic validation with actual UNSW-NB15 flows replayed through the network stack; (2) evaluation on modern datasets (CIC-IDS2017, CSE-CIC-IDS2018) capturing contemporary threats; (3) expert evaluation of explanation quality; (4) adversarial robustness testing against evasion techniques; and (5) cost analysis with actual API pricing. Technical improvements involve migrating to scalable vector databases (Qdrant, Milvus, Weaviate), implementing advanced RAG techniques (hybrid search, re-ranking), and enterprise SIEM integration. Implementation code and evaluation scripts will be released to facilitate reproducibility.

References

[1] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network Intrusion Detection for IoT Security Based on Learning Techniques," IEEE Communications Surveys & Tutorials, vol. 21, no. 3, pp. 2671–2701, 2019.

[2] A. L. Buczak and E. Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," IEEE Communications Surveys & Tutorials, vol. 18, no. 2, pp. 1153–1176, 2016.

[3] N. Sultana, N. Chilamkurti, W. Peng, and R. Alhadad, "Survey on SDN Based Network Intrusion Detection System Using Machine Learning Approaches," Peer-to-Peer Networking and Applications, vol. 12, no. 2, pp. 493–501, 2019.

[4] M. Brundage et al., "The Malicious Use of Artificial Intelligence: Forecasting, Prevention, and Mitigation," arXiv preprint arXiv:1802.07228, 2018.

[5] M. Du, F. Li, G. Zheng, and V. Srikumar, "DeepLog: Anomaly Detection and Diagnosis from System Logs Through Deep Learning," in Proc. ACM SIGSAC Conf., 2019, pp. 1285–1298.

[6] M. Tserenkhuu, M. D. Hossain, Y. Taenaka, and Y. Kadobayashi, "Intrusion Detection System Framework for SDN-Based IoT Networks Using Deep Learning Approaches With XAI-Based Feature Selection Techniques and Domain-Constrained Features," IEEE Access, vol. 13, pp. 136864–136880, 2025.

[7] A. B. Arrieta et al., "Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges," Information Fusion, vol. 58, pp. 82–115, 2020.

[8] T. Brown et al., "Language Models are Few-Shot Learners," in Advances in Neural Information Processing Systems, vol. 33, 2020, pp. 1877–1901.

[9] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proc. NAACL-HLT, 2019, pp. 4171–4186.

[10] P. R. B. Houssel, P. Singh, S. Layeghy, and M. Portmann, "Towards Explainable Network Intrusion Detection Using Large Language Models," in 2024 IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT), 2024.

[11] Y. Farrukh, S. Wali, I. Khan, and N. Bastian, "XG-NID: Dual-Modality Network Intrusion Detection using a Heterogeneous Graph Neural Network and Large Language Model," arXiv preprint arXiv:2408.16021, 2024.

[12] J. Zhang and M. Zulkernine, "Network Intrusion Detection Using Random Forests," in Third Annual Conf. Privacy, Security and Trust, 2008.

[13] O. Arreche, T. Guntur, and M. Abdallah, "XAI-based Feature Selection for Improved Network Intrusion Detection Systems," arXiv preprint arXiv:2410.10050, 2024.

[14] K. Ubaidillah, S. Hisham, F. Ernawan, G. Badshah, and E. Suharto, "Intrusion Detection System using Autoencoder-based Deep Neural Network for SME Cybersecurity," in Proc. Int. Conf. on Information and Communication Systems (ICICoS), 2021, pp. 210–215.

[15] Z. Lu, H. Xu, and J. Pan, "Research on Intrusion Detection Based on Self-Attention and Residual Network," in Proc. Int. Conf. on Intelligent Automation, Electronics and Communication Systems (IAECST), 2024, pp. 420–423.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," arXiv preprint arXiv:1706.03762, 2017.

[17] A.-R. El Mehdi Baahmed, G. Andresini, C. Robardet, and A. Appice, "Using Graph Neural Networks for the Detection and Explanation of Network Intrusions," in Proc. ECML PKDD 2023 Workshops, 2023, pp. 201–216.

[18] A. Radford, J. Wu, R. Child, et al., "Language Models are Unsupervised Multitask Learners," OpenAI Blog, vol. 1, no. 8, p. 9, 2019.

[19] K. Sauka, G.-Y. Shin, D.-W. Kim, and M.-M. Han, "Adversarial Robust and Explainable Network Intrusion Detection Systems Based on Deep Learning," Applied Sciences, vol. 12, 2022.

[20] N. Moustafa and J. Slay, "UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems," in Proc. Military Communications and Information Systems Conference (MilCIS), 2015, pp. 1–6