2026 International Conference on Computing, Networking and Communications (ICNC) | 979-8-3315-7111-5/26/$31.00 ©2026 IEEE | DOI: 10.1109/ICNC68183.2026.11416826

ReGAIN: Retrieval-Grounded AI Framework for Network Traffic Analysis

Shaghayegh Shajarian
Computer Science
North Carolina A&T State University
Greensboro NC, USA
0009-0003-7334-3864

Kennedy Marsh
Computer Science
North Carolina A&T State University
Greensboro NC, USA
0000-0002-5987-252X

James Benson
Institute for Cyber Security
University of Texas at San Antonio
San Antonio TX, USA
0000-0001-7209-2344

Sajad Khorsandroo
Computer Science
North Carolina A&T State University
Greensboro NC, USA
0000-0003-0649-9247

Mahmoud Abdelsalam
Computer Science
North Carolina A&T State University
Greensboro NC, USA
0000-0001-5627-5239

***Abstract*—Modern networks generate vast, heterogeneous traffic that must be continuously analyzed for security and performance. Traditional network traffic analysis systems, whether rule-based or machine learning–driven, often suffer from high false positives and lack interpretability, limiting analyst trust. In this paper, we present ReGAIN, a multi-stage framework that combines traffic summarization, retrieval-augmented generation (RAG), and Large Language Model (LLM) reasoning for transparent and accurate network traffic analysis. ReGAIN creates natural-language summaries from network traffic, embeds them into a multi-collection vector database, and utilizes a hierarchical retrieval pipeline to ground LLM responses with evidence citations. The pipeline features metadata-based filtering, MMR sampling, a two-stage cross-encoder reranking mechanism, and an abstention mechanism to reduce hallucinations and ensure grounded reasoning. Evaluated on ICMP ping flood and TCP SYN flood traces from the real-world traffic dataset, it demonstrates robust performance, achieving accuracy between 95.95% and 98.82% across different attack types and evaluation benchmarks. These results are validated against two complementary sources: dataset ground truth and human expert assessments. ReGAIN also outperforms rule-based, classical ML, and deep learning baselines while providing unique explainability through trustworthy, verifiable responses.**

***Index Terms*—Network Traffic Analysis, Large Language Models (LLMs), Retrieval-Augmented Generation (RAG), Intelligent Networks, Network Security.**

I. Introduction

Modern networks generate massive volumes of traffic that must be continuously monitored for performance, reliability, and security. Alongside real-time monitoring, historical traffic data (e.g, packet captures (PCAPs) and flow records) is invaluable for forensic investigations. These records often contain the clearest evidence of malicious activity, revealing abnormal payloads, tunneling techniques, or credential theft. Retrospective traffic analysis thus plays a critical role in reconstructing attack campaigns, validating alerts, and improving defensive strategies. However, traditional network traffic analysis systems face several limitations. Rule-based systems (e.g., Snort, Suricata) rely on manually crafted signatures, which require constant maintenance, produce high false positive rates, and offer limited explanations. Machine learning approaches, such as Support Vector Machines (SVMs), Random Forests, and deep learning models, achieve strong detection accuracy but often operate as black boxes. This lack of explainability reduces analyst trust and complicates incident response, as operators must manually correlate alerts with supporting evidence from multiple data sources. Large Language Models (LLMs) have shown promise in network operations [1], [2], with reasoning over semi-structured data, and generating human-readable insights. However, in purely generative modes, they risk hallucinations and unverifiable claims. Retrieval-Augmented Generation (RAG) mitigates this by grounding LLM outputs in external knowledge sources, ensuring that generated explanations are supported by verifiable evidence [3].

In this paper, we present ReGAIN (Retrieval-Grounded AI for Network Traffic Analysis), a multi-stage, LLM-driven framework that integrates hierarchical semantic retrieval, evidence quality monitoring, and citation-backed reasoning. Building on our previous work [4], ReGAIN comprises four primary components (detailed in Section III): (1) a data ingestion pipeline that transforms heterogeneous network telemetry into natural language summaries, (2) a multi-collection vector knowledge base that semantically indexes these summaries with structured metadata for efficient retrieval, (3) a retrieval-augmented reasoning engine that combines multiple techniques (e.g., metadata filtering, cross-encoder reranking) to select high-quality evidence, and (4) an LLM-driven analysis component that generates human readable explanations with explicit citations to supporting records. Unlike traditional RAG systems that rely on a single knowledge base and lack retrieval quality controls, ReGAIN leverages multi-collection retrieval across specialized knowledge bases, adaptive context selec-

Code, data, and model availability: https://github.com/270771/llm-traffic tion via automated metadata filtering and Maximal Marginal Relevance (MMR) diversity sampling, multi-stage retrieval refinement using bi-encoder search followed by cross-encoder reranking, and an abstention mechanism that returns diagnostic feedback when retrieval quality is insufficient, preventing hallucinations while providing human readable explanations with explicit citations to supporting records. The contributions of this work are as follows:

- We introduce the ReGAIN framework, which unifies structured traffic representation, semantic embedding, vector retrieval, and LLM reasoning for network traffic analysis.
- We design a pipeline with multi-collection retrieval, adaptive context selection, multi-stage reranking, and an abstention mechanism that mitigates hallucinations.
- We conduct a dual-mode evaluation, combining automated and expert-based approaches, on a real-world network traffic dataset that includes ICMP ping flood and TCP SYN flood attacks, illustrating strong detection performance across both scenarios.
- We benchmark ReGAIN against traditional rule-based, classical machine learning, and deep learning approaches, demonstrating superior performance.

II. Related Work

Recent studies have explored how LLMs can support networking tasks beyond traditional classifiers. NetLLM [5] adapts LLMs for a range of networking problems and represents an early step toward unified LLM-driven workflows in this area. ShieldGPT [6] applies an LLM-driven approach to detect and mitigate DDoS attacks, showing how language-based reasoning can complement existing traffic analysis tools. In network security, several works emphasize that LLMs are most useful for generating explanations that help operators understand alerts rather than replacing detection systems completely. Houssel *et al.* evaluate LLMs as explainable components for intrusion detection and later propose eX-NIDS, which focuses on improving interpretability for flow-based NIDSs [7], [8]. TrafficLLM [9] introduces domain-specific traffic representations and dual-stage fine-tuning to improve generalization across different network traffic tasks.

RAG has also started to gain attention in cybersecurity. Rahman *et al.* show that combining knowledge graphs with RAG can improve cyber threat analysis by linking model outputs to structured information. Other works utilize retrieval-enhanced LLMs to support incident response and decision making [10], [11]. Meanwhile, the broader security community has begun benchmarking and evaluating LLMs for security related applications, highlighting the need for domain-specific evaluation frameworks [12]. Despite these advances, most existing LLM-driven systems focus on either detection accuracy or interpretability in isolation. To fill in this gap, ReGAIN contains retrieval-augmented reasoning that grounds LLM outputs in cited traffic evidence and related artifacts, a deterministic summarization layer that converts raw network logs into concise natural language descriptions for embedding, a hybrid evaluation strategy that combines ground truth comparison with human expert labels (see Table I).

TABLE I
Comparison of ReGAIN with related LLM-based systems.

Framework	RAG	Traffic Repr.	Expert Val.	Evidence Cited
NetLLM [5]	✗	Structured features (task adapters)	✗	Limited
ShieldGPT [6]	✗	Flow-level inputs	✗	Generative
eX-NIDS [8]	✗	Flow inputs + templates	✗	Template
TrafficLLM [9]	✗	Generic traffic repr.	✗	Domain prompts
ReGAIN (ours)	✓	**NL summaries + embed.**	✓	**Cited evidence**

III. ReGAIN Framework

Our proposed framework, ReGAIN, as illustrated in the Figure 1, comprises four main components: (1) data ingestion and summarization, (2) semantic vectorization and knowledge base builder, (3) retrieval-augmented reasoning and generation, and (4) human-in-the-loop interaction.

A. Data Ingestion and Summarization

Network traffic telemetry originates from different sources, including log files, CSV anomaly annotations, and flow records. To enable uniform downstream processing, these inputs are normalized into a structured schema:

$$r_i = \{ts_i, src_i, dst_i, p_i, proto_i, \ell_i\}, \tag{1}$$

where ts_i is the timestamp, src_i and dst_i are the source and destination IPs, p_i is the port, $proto_i$ is the protocol, and ℓ_i is the anomaly label. Each record is transformed into a natural-language summary:

$$s_i = f_{\text{sum}}(r_i), \tag{2}$$

where f_{sum} is a deterministic summarization function.

For instance, a record such as `2024-08-15 10:05:23, 192.0.2.7, 203.0.113.5, icmp, label=DoS` is summarized as: *"At 10:05:23 on August 15, 2024, host 192.0.2.7 sent an ICMP request to 203.0.113.5, flagged as a potential DoS anomaly."*.

This summarization serves several purposes. First, it reduces structured telemetry into an information-dense, human readable representation, preventing LLM context windows from being saturated by noise from raw logs. Second, it exposes network semantics (endpoints, protocol, timing, labels) in natural language, which improves the quality of embeddings and facilitates meaningful retrieval. Finally, providing concise, interpretable summaries ensures that when the ReGAIN cites supporting records as evidence, they are transparent and verifiable.

B. Semantic Vectorization and Knowledge Base Builder

Each natural-language summary is encoded into a d-dimensional embedding:

$$v_i = f_{\text{embed}}(s_i) \in \mathbb{R}^d, \tag{3}$$

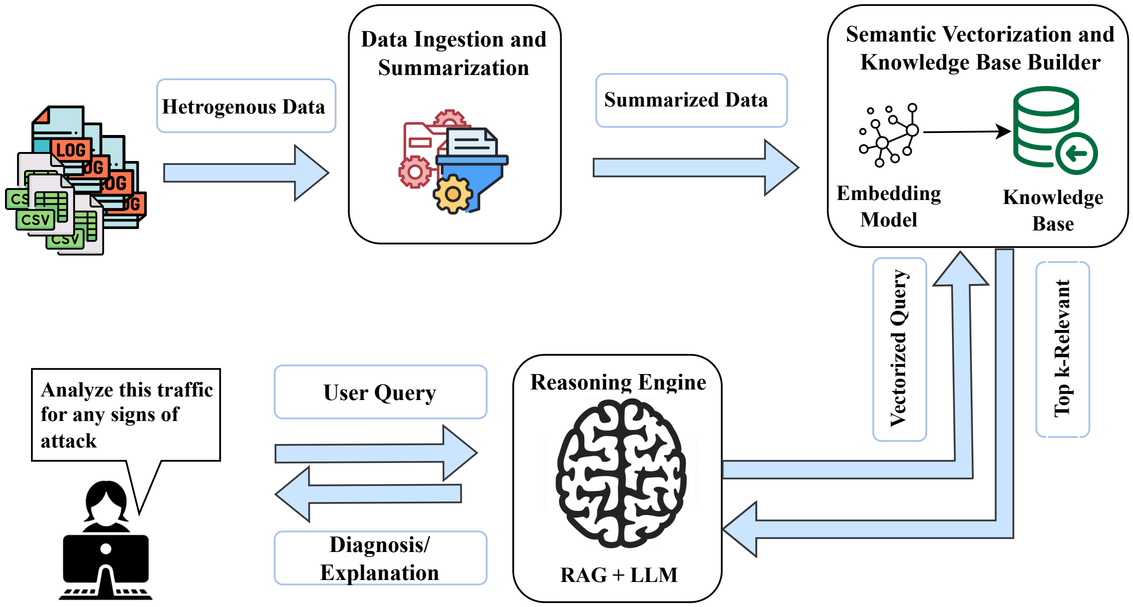


Fig. 1. ReGAIN architecture: pipeline from traffic ingestion to reasoning.

where f_{embed} denotes a transformer-based embedding model. Each knowledge base entry is stored as:

$$e_i = (s_i, v_i, m_i), \tag{4}$$

with m_i containing structured metadata derived from the *5-tuple* $(src\_IP, dst\_IP, src\_port, dst\_port, protocol)$, along with entry labels, and timestamps. To improve retrieval precision and context diversity, ReGAIN employs a *multi-collection architecture* comprising three specialized vector databases: a telemetry collection containing enriched flow-level and packet-level summaries derived from PCAPs and log files, an anomaly collection capturing labeled or auto-detected attack instances with metadata, and a heuristic collection containing reference material such as detection heuristics and post incident annotations. Each collection is semantically indexed but remains logically isolated, allowing ReGAIN to retrieve context selectively or in parallel depending on query intent. For non-telemetry artifacts such as RFCs or incident tickets, the same representation applies: the document passage is treated as s_i, embedded into v_i, and tagged with relevant m_i metadata.

C. Retrieval-Augmented Reasoning and Generation

When an analyst issues a query q, it is embedded into the same vector space as the corpus:

$$v_q = f_{\text{embed}}(q). \tag{5}$$

Similarity is computed via cosine similarity:

$$\text{sim}(v_q, v_i) = \frac{v_q \cdot v_i}{\|v_q\| \, \|v_i\|}. \tag{6}$$

To improve fidelity and prevent hallucination, we adopt a *hierarchical retrieval strategy* that combines metadata-aware filtering with multi-stage semantic search. When a query is received, named entity and IP extraction automatically identify relevant metadata (e.g., destination IPs, protocols, ports, timestamps). These elements are used to construct a filter ϕ applied across all collections, narrowing the search to relevant flows or anomaly categories. Candidates satisfying ϕ are retrieved and ranked by semantic similarity:

$$R_\phi(q) = \{e_i \in \mathcal{E}_\phi \mid \text{sim}(v_q, v_i) \geq \tau\}. \tag{7}$$

where τ serves as the similarity threshold below which candidates are discarded.

To reduce redundancy, *MMR* is applied to select a diverse subset that balances relevance and coverage, ensuring the LLM receives complementary evidence from telemetry, anomaly, and heuristic collections. These MMR-pruned candidates are then reranked hierarchically using a *bi-encoder* followed by a *cross-encoder*. The bi-encoder captures coarse semantic alignment, while the cross-encoder refines context sensitivity between the query and evidence pairs (q, e_i).

Moreover, before generation, an *abstention mechanism* implements a pre-generation quality gate to assess the coherence of retrieved evidence. If the number of high-confidence items $|R'_q|$ falls below a predefined threshold, the framework abstains from generating a response and returns undecidable, citing missing or inconsistent evidence. The top-k results that pass quality checks are passed to the LLM:

$$y_q = f_{\text{LLM}}(q, R'_q), \tag{8}$$

The output, y_q, follows a structured schema: a verdict (attack, no-attack, undecidable), the evidence and reasoning

TABLE II
EXPERIMENTAL ENVIRONMENT AND TOOLING

Component	Configuration
Data source	MAWILab v1.1 PCAPs (Jan 2022)
Parsing	Structured connection logs
Embeddings	all-MiniLM-L6-v2 (384-D)
Cross-encoder	cross-encoder/ms-marco-MiniLM-L-6-v2
Vector store	ChromaDB with three persistent collections
Orchestration	LangChain framework
LLM	GPT-4.1-nano, temperature = 0
Similarity threshold	$\tau = 0.3$
MMR parameters	k = 3–6, fetch_k=$3k$

behind, and one or two recommended mitigations. If the similarity scores fall below a threshold τ or the evidence is inconsistent, the system abstains by outputting “undecidable” and listing missing context.

D. Human-in-the-Loop Interaction

The framework is designed as a decision-support tool, enabling network analysts to iteratively refine their investigations. Analysts can reformulate queries based on results:

$$q^{(t+1)} = g(q^{(t)}, y_q), \tag{9}$$

where g is an analyst-driven reformulation function.

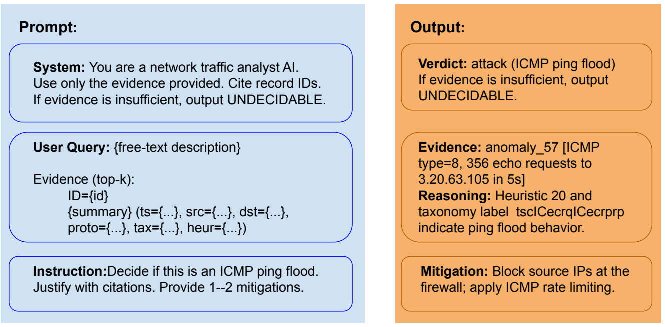
Example: An analyst may begin with a broad query (*“Show anomalies involving 203.0.113.5”*), receive evidence of ICMP floods, and refine to a narrower one (*“Compare with TCP SYN activity in the same interval”*). The framework maintains context across iterations, supporting forensic reasoning workflows that mimic real-world incident response.

IV. EXPERIMENTAL SETUP

This section describes the dataset, tools, and configurations used to implement and evaluate our framework. The software stack is summarized in Table II.

a) Dataset and Knowledge Base: We use the MAWILab v1.1 network traces [13]. For each day, MAWILab provides (i) PCAPs and (ii) structured anomaly CSVs with two complementary labels, comprising *heuristic* (signature/flag/port/type–code driven) and *taxonomy* (behavioral categories such as DoS, scans, tunneling). Each row includes the 5-tuple (where applicable), heuristic/taxonomy codes, severity (anomalous, suspicious, notice, benign), and identifiers. In this study, we focus on analyzing ICMP and TCP network activities, specifically ping flood and SYN flood attacks. We selected raw MAWILab PCAPs captured on January 1, 9, and 10, 2022. These captures reflect contemporary Internet conditions and protocol distributions, and also retain the detailed anomaly annotations required for reproducible evaluation. We store the embeddings and metadata in ChromaDB [14], an open-source vector database designed for high-dimensional similarity search.

b) Prompt Structure: The prompt template controls how retrieved evidence is presented to the language model. Our design goal was to balance three requirements: (i) grounding the model’s reasoning in verifiable evidence, (ii) enforcing a consistent and auditable output format, and (iii) ensuring actionable recommendations for operators. The system instruction requires the model to cite retrieved record IDs or heuristic codes in its reasoning. If the retrieval context is insufficient, the model is instructed to output the keyword *undecidable* and list the missing evidence. The response schema follows a three-part structure, incorporating an alert summary describing the detected activity, a justification citing retrieved evidence, and one or two concise mitigation steps. The prompt explicitly instructs the model to provide assertive, confident assessments and avoid hedging language (e.g., “might”, “possibly”), which makes outputs more actionable rather than tentative. In the deployed environment, the prompt and model output are displayed through a lightweight command-line interface (CLI). This interface enables analysts to inspect retrieved evidence, view the structured LLM response, and iteratively refine their queries. Figure 2 demonstrates an abridged prompt and output of the framework.



System: You are a network traffic analyst AI.
Use only the evidence provided. Cite record IDs.
If evidence is insufficient, output UNDECIDABLE.

User Query: {free-text description}

Evidence (top-k):
ID={id}
{summary} (ts={...}, src={...}, dst={...},
proto={...}, tax={...}, heur={...})

Instruction: Decide if this is an ICMP ping flood.
Justify with citations. Provide 1--2 mitigations.

Verdict: attack (ICMP ping flood)
If evidence is insufficient, output
UNDECIDABLE.

Evidence: anomaly_57 [ICMP
type=8, 356 echo requests to
3.20.63.105 in 5s]
Reasoning: Heuristic 20 and
taxonomy label tscICecrqICecrprp
indicate ping flood behavior.

Mitigation: Block source IPs at the
firewall; apply ICMP rate limiting.

Fig. 2. An abridged prompt and output of the system.

c) Inference Parameters: To ensure the comparability of results, we use a uniform instruction block across all runs. We also employ deterministic decoding (temperature $\approx$ 0) to minimize variability in the LLM outputs. We cap the retrieved evidence at $k \in \{3, 5\}$: a smaller k reduces context dilution and enforces concise reasoning, while a larger k provides additional corroborating records. This range was selected empirically as a balance between precision (avoiding irrelevant context) and recall (ensuring sufficient evidence is available).

V. EVALUATION

We evaluated the ReGAIN framework using two complementary methodologies across two attack scenarios, TCP SYN floods and ICMP ping floods. The first method involved an automated comparison against ground-truth annotations, and the second one incorporated manual expert judgment to assess the framework’s performance independently. Table III summarizes key performance metrics for SYN and Ping flood attacks under both ground-truth and expert labels. Performance was evaluated using standard metrics. Accuracy was computed as $\frac{TP+TN}{TP+TN+FP+FN}$; Precision as $\frac{TP}{TP+FP}$; Recall as $\frac{TP}{TP+FN}$; and the F1 score as $2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$.

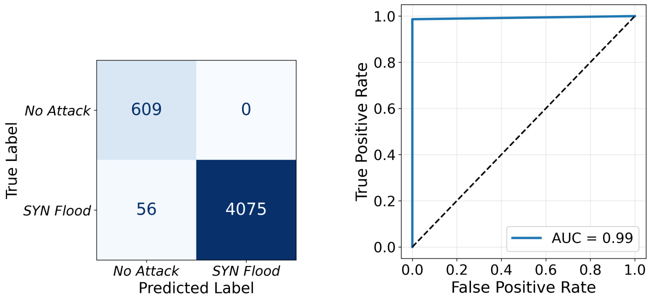
To generate ground truth labels, SYN flood attacks were identified using TCP connection states in the logs. Connections marked as `S0` (SYN sent, no reply) indicate incomplete handshakes typical of SYN floods, and those marked as (`SH`, `SF`, `RSTR`, `RSTO`, `OTH`) reflect normal or benign traffic. For ping flood attacks, a sliding-window detector flagged logs with ten or more ICMP echo requests (type 8) from the same source to the same destination within a 20-second window as attacks, capturing both one-to-many and many-to-one flooding behavior. These automated labels were cross-validated with MAWILab's heuristic-20 anomalies to improve labeling accuracy.

To establish an independent validation baseline, a subset of connection logs underwent blind expert adjudication based on traffic characteristics and domain knowledge. Experts analyzed Zeek logs using known attack signatures. For SYN flood detection, experts assessed TCP connection states, labeling events with high volumes of incomplete handshakes (`S0`) from specific IPs as attacks. They differentiated attacks from benign failures by considering connection rate, IP diversity, timing, and correlation with MAWILab anomalies. For ping flood detection, ICMP traffic was examined based on type codes, IP pairs, and timing. Events were labeled as attacks if MAWILab listed the IPs with heuristic code 20, or ten or more ICMP echo requests from a single source to a destination occurred in a short time frame. Logs with 5–9 requests were flagged for further review, while those with fewer than five were deemed benign.

A. SYN Flood Attack

a) Results Against Ground Truth: As illustrated in Figure 3a, the confusion matrix reveals a highly favorable distribution with 4,075 true positives, 609 true negatives, zero false positives, and only 56 false negatives. This corresponds to an overall accuracy of 98.82%, precision of 100.00%, recall of 98.64%, and an F1 score of 99.32%. The ROC curve in Figure 3b further illustrates this discriminative capability, yielding an AUC of 0.99. The perfect precision indicates that when ReGAIN identifies SYN flood activity, it does so with complete certainty, while the near-perfect recall demonstrates that very few actual attacks escape detection.

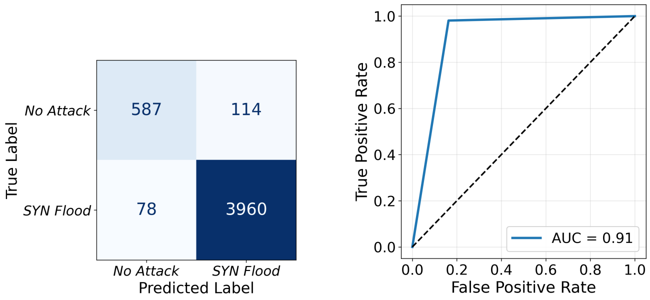
b) Results Against Expert Labels: Expert evaluation of the SYN flood attack revealed a different pattern of refinement, where the confusion matrix (Figure 4a) shows 3,960 true positives, 587 true negatives, 114 false positives, and 78 false negatives. This represents a shift from the ground-truth evaluation, showing that although precision remained exceptionally high at 97.20%, recall decreased slightly to 98.07%, resulting in an overall accuracy of 95.95% and an F1 score of 97.63%. The ROC curve in Figure 4b exhibits continued strong discriminative performance with an AUC of 0.98. The emergence of false positives and false negatives under expert review suggests that some automated ground-truth labels may have been overly permissive, and that certain edge cases, such as partial SYN floods or rate-limited attack attempts, require human judgment to classify accurately.



(a) Confusion Matrix

(b) ROC Curve

Fig. 3. SYN Flood Evaluation Against Ground Truth: (a) Confusion matrix, (b) ROC curve.



(a) Confusion Matrix

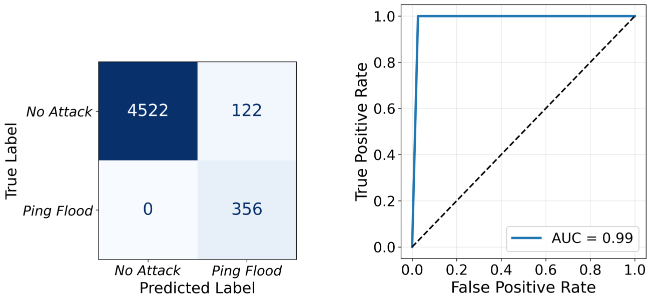
(b) ROC Curve

Fig. 4. SYN Flood Evaluation Against Expert Labels: (a) Confusion matrix, (b) ROC curve.

B. Ping Flood Attack

a) Results Against Ground Truth: For the ping flood attack, the system achieved perfect recall (100.00%) while identifying all 356 true attack instances with zero false negatives, as shown in Figure 5a. This perfect sensitivity came at the cost of precision, which measured 74.48% due to 122 false positives, benign ICMP traffic misclassified as attacks. The overall accuracy reached 97.56% with an F1 score of 85.37%. The ROC curve in Figure 5b demonstrates strong discriminative capability with an AUC of 0.99, indicating near-perfect class separability despite the precision trade-off. These results suggest that although the system maintains exceptional recall for ping flood attacks, its precision is affected by the similarity between benign diagnostic ICMP activity and actual flood patterns.

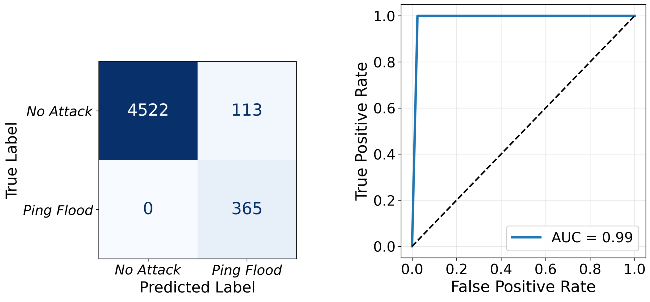
b) Results Against Expert Labels: In expert evaluation, the ping flood attack maintained perfect recall (100.00%), correctly identifying all 365 validated attack instances without any false negatives, as shown in Figure 6a. Compared to the ground-truth evaluation, the expert labels identified nine additional true positive cases (365 vs. 356), representing instances where the framework correctly detected attacks that were missing or underrepresented in the original automated annotations. The number of false positives decreased modestly from 122 to 113, which increased the precision to 76.36% and improved overall accuracy to 97.74%. The F1 score increased



(a) Confusion Matrix

(b) ROC Curve

Fig. 5. Ping Flood Evaluation Against Ground Truth: (a) Confusion matrix, (b) ROC curve.



(a) Confusion Matrix

(b) ROC Curve

Fig. 6. Ping Flood Evaluation Against Expert Labels: (a) Confusion matrix, (b) ROC curve.

to 86.60%, and the ROC curve in Figure 6b maintained an AUC of 0.99.

The reduced precision in our results shows that, despite ReGAIN's perfect recall in both evaluations, it has a tendency to overclassify certain benign ICMP activities as attacks. Most false positives originated from short-lived or diagnostic ICMP bursts, such as network reachability checks, latency probes, or automated monitoring tasks, that temporarily showed traffic patterns similar to genuine ping floods. Since MAWILab annotations do not always distinguish between benign high-frequency ICMP traffic and attack-induced floods, the framework conservatively labeled these cases as anomalous.

C. Comparison with Baseline Methods

To evaluate the effectiveness of ReGAIN, we conducted a comparative analysis against five baseline detection approaches: (1) traditional rule-based intrusion detection employing Snort-style threshold heuristics, (2) SVM with Radial Basis Function kernels, (3) Random Forest ensemble classifiers configured with 100 decision trees, (4) one-dimensional Convolutional Neural Networks (CNN), and (5) two-layer Long Short-Term Memory (LSTM) networks. All models use 30 numerical features extracted from Zeek conn.log files (protocol ratios, connection states, temporal stats, byte/packet volumes, IP/port diversity) and these features are normalized using StandardScaler (z-score). Also, all models were trained and evaluated on identical partitions of the MAWILab dataset to ensure methodological consistency, with performance metrics calculated using ground-truth labels.

TABLE III
PERFORMANCE METRICS FOR SYN AND PING FLOOD ATTACKS

Attack	Acc.	Prec.	Recall	F1	AUC
SYN (GT)	98.82%	100.00%	98.64%	99.32%	0.99
SYN (Expert)	95.95%	97.20%	98.07%	97.63%	0.98
Ping (GT)	97.56%	74.48%	100.00%	85.37%	0.99
Ping (Expert)	97.74%	76.36%	100.00%	86.60%	0.99

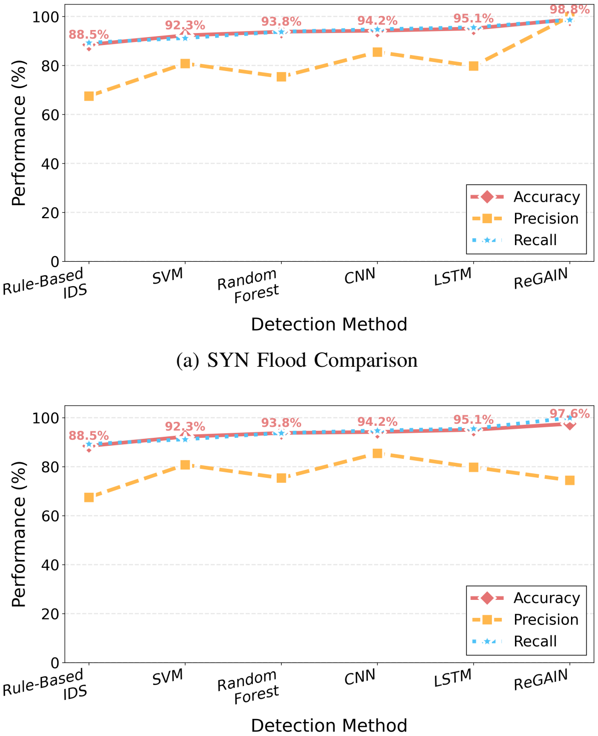
Figure 7 presents a comparative analysis of ReGAIN and the baselines. For SYN Flood attack (Figure 7a), compared to the best-performing baseline (LSTM), ReGAIN improves accuracy by 3.7 percentage points, precision by 14.5 points, and recall by 3.8 points. This balanced performance profile demonstrates ReGAIN's ability to maintain high detection sensitivity and simultaneously minimize false positives, which is a challenging combination for conventional machine learning approaches.

For the Ping Flood attack (Figure 7b), ReGAIN achieves superior overall performance, surpassing the strongest baseline (LSTM). Most notably, ReGAIN achieves perfect recall (100.0% vs. 95.6% for LSTM), ensuring zero false negatives at the cost of reduced precision (74.5%). This precision-recall trade-off reflects a deliberate design choice that prioritizes detection sensitivity, i.e., a critical requirement in network security applications where failing to detect attacks carries substantially higher risk than investigating false alarms.

Beyond quantitative performance improvements, ReGAIN offers critical qualitative advantages that distinguish it from traditional detection systems. ReGAIN generates human-readable natural language explanations for each detection decision. This interpretability addresses a fundamental limitation in network security operations, where security analysts require actionable insights rather than opaque verdicts. Furthermore, ReGAIN's conversational interface enables interactive refinement of detection criteria and iterative querying of network behavior, facilitating collaborative human-AI investigation workflows that are infeasible with static classification models. These capabilities transform network traffic analysis from a passive alerting mechanism into an interactive analytical tool, supporting both automated detection and human-guided threat hunting activities.

VI. CONCLUSION AND FUTURE DIRECTIONS

This work presented ReGAIN, a novel framework that integrates network traffic summarization, semantic search, and LLM-driven reasoning to support transparent and accurate traffic analysis. By transforming raw network data into descriptive, embedded summaries stored in a multi-collection vector database, ReGAIN enables efficient, evidence-grounded retrieval of relevant historical patterns. The system's hierarchical retrieval and re-ranking mechanisms, combined with metadata filtering and an abstention strategy, help mitigate



(a) SYN Flood Comparison

(b) Ping Flood Comparison

Fig. 7. Performance comparison of ReGAIN against baseline detectors for Ping Flood and SYN Flood attacks.

hallucinations and ensure interpretability. Evaluations of ICMP ping flood and TCP SYN flood scenarios, containing 10,000 labeled instances, demonstrated ReGAIN's strong performance, with high accuracy (95.95–98.82%) and near-perfect recall (98.64–100%), outperforming traditional rule-based and learning-based baselines. These results show the promise of LLM-driven reasoning and retrieval augmentation for network traffic analysis.

There are several limitations and areas for improvement that we aim to address in future work. Although ReGAIN demonstrates strong performance, it relies on GPT-4-class models accessed via remote APIs and utilizes lightweight 384-dimensional embeddings designed to favor reasoning quality over computational throughput. As a result, inference latency is constrained by API communication, making ReGAIN more appropriate for retrospective analysis and forensic investigations rather than real-time monitoring. To reduce latency and enhance data privacy, future deployments can incorporate on-premise models such as LLaMA-3-8B, Mistral-7B, or Phi-3, and local embedding models to enable fully offline operation in air-gapped or sensitive environments.

We also note a precision gap (approximately 74–76%) in detecting ping flood attacks, primarily due to benign ICMP activity that can resemble attack patterns. To improve detection accuracy, we plan to consider dynamic similarity thresholds based on protocol type and traffic volume, as well as temporal and rate-based filters to differentiate short bursts from sustained attacks. In addition, future work will focus on evaluating the clarity and effectiveness of the generated natural language outputs, ensuring that summaries and explanations are not only accurate but also easily interpretable by analysts. These directions will strengthen ReGAIN as a practical, lightweight, and explainable framework for modern network traffic analysis.

Acknowledgment

This work is partially supported by NSF grants 2113945, 2200538, 2416992, and 2230610 at NC A&T SU. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding agency.

References

[1] Boateng, Gordon Owusu, Hani Sami, Ahmed Alagha, Hanae Elmekki, Ahmad Hammoud, Rabeb Mizouni, Azzam Mourad et al. ”A survey on large language models for communication, network, and service management: Application insights, challenges, and future directions.” IEEE Communications Surveys & Tutorials (2025).

[2] Shaghayegh Shajarian, Sajad Khorsandroo, Mahmoud Abdelsalam. Self-Running Networks: A Comprehensive Survey of Foundations, Applications, and Challenges. TechRxiv. July 22, 2025.

[3] Lewis, Patrick, et al. ”Retrieval-augmented generation for knowledge-intensive nlp tasks.” Advances in Neural Information Processing Systems 33 (2020): 9459-9474.

[4] Shajarian, Shaghayegh, Sajad Khorsandroo, and Mahmoud Abdelsalam. ”Poster: Intelligent Network Management: RAG-Enhanced LLMs for Log Analysis, Troubleshooting, and Documentation.” Proceedings of the 20th International Conference on emerging Networking EXperiments and Technologies. 2024.

[5] Wu, Duo, Xianda Wang, Yaqi Qiao, Zhi Wang, Junchen Jiang, Shuguang Cui, and Fangxin Wang. ”Netllm: Adapting large language models for networking.” In Proceedings of the ACM SIGCOMM 2024 Conference, pp. 661-678. 2024.

[6] ZWang, Tongze, Xiaohui Xie, Lei Zhang, Chuyi Wang, Liang Zhang, and Yong Cui. ”Shieldgpt: An llm-based framework for ddos mitigation.” In Proceedings of the 8th asia-pacific workshop on networking, pp. 108-114. 2024.

[7] Houssel, Paul RB, Priyanka Singh, Siamak Layeghy, and Marius Portmann. ”Towards explainable network intrusion detection using large language models.” In 2024 IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT), pp. 67-72. IEEE, 2024.

[8] Houssel, Paul RB, Siamak Layeghy, Priyanka Singh, and Marius Portmann. ”eX-NIDS: A Framework for Explainable Network Intrusion Detection Leveraging Large Language Models.” arXiv preprint arXiv:2507.16241 (2025).

[9] Cui, Tianyu, Xinjie Lin, Sijia Li, Miao Chen, Qilei Yin, Qi Li, and Ke Xu. ”Trafficllm: Enhancing large language models for network traffic analysis with generic traffic representation.” arXiv preprint arXiv:2504.04222 (2025).

[10] Rahman, Moqsadur, Krish O. Piryani, Aaron M. Sanchez, Sai Munikoti, Luis De La Torre, Maxwell S. Levin, Monika Akbar, Mahmud Hossain, Monowar Hasan, and Mahantesh Halappanavar. Retrieval augmented generation for robust cyber defense. No. PNNL-36792. Pacific Northwest National Laboratory (PNNL), Richland, WA (United States), 2024.

[11] Hammar, Kim, Tansu Alpcan, and Emil C. Lupu. ”Incident Response Planning Using a Lightweight Large Language Model with Reduced Hallucination.” arXiv preprint arXiv:2508.05188 (2025).

[12] Lin, Jie, and David Mohaisen. ”From large to mammoth: A comparative evaluation of large language models in vulnerability detection.” In Proceedings of the 2025 Network and Distributed System Security Symposium (NDSS). 2025.

[13] R. Fontugne, P. Borgnat, P. Abry, K. Fukuda. ”MAWILab: Combining diverse anomaly detectors for automated anomaly labeling and performance benchmarking”. ACM CoNEXT 2010. Philadelphia, PA. December 2010

[14] chroma-core. 2025. chroma. GitHub repository. Accessed September 12. https://github.com/chroma-core/chroma.