RAG-I: Retrieval-Augmented Generative Intelligence for Adaptive DDoS Detection in Evolving Networks

Ritisha Duggempudi
B.Tech Final Year
Roll No: 2211CS020139

Kotha Harshitha Vijaya Sri
B.Tech Final Year
Roll No: 2211CS020265

Mokirala Akhil
B.Tech Final Year
Roll No: 2211CS020330

K. Venkata Kamalesh
B.Tech Final Year
Roll No: 2211CS020258

Under the guidance of

G. Dileep Kumar
Assistant Professor
Department of Artificial Intelligence and Machine Learning
Malla Reddy University, Hyderabad, Telangana, India
dileepkumar.g@mallareddyuniversity.ac.in

Abstract—Network infrastructures are increasingly vulnerable to Distributed Denial-of-Service (DDoS) attacks as adversaries continuously evolve their attack strategies [1]. Existing detection mechanisms suffer from three major limitations: inability to recognize evolving attack patterns within ongoing campaigns, excessive false-alarm rates that burden security analysts, and lack of interpretable explanations for flagged traffic [6]. This study introduces RAG-I (Retrieval-Augmented Generative Intelligence), a framework that leverages the SEER-DD algorithm to address these deficiencies. Our research methodology involved systematic experimentation with four baseline approaches— Isolation Forest, One-Class SVM, TiDE (Temporal Intent Drift Estimator), and Deep Autoencoder—each revealing specific shortcomings that informed SEER-DD's architecture. The framework incorporates five technical innovations: (i) transformation of numerical traffic statistics into semantic textual representations; (ii) temporal monitoring of attack behavior evolution; (iii) identification of previously unseen attack patterns through similarity-based anomaly analysis; (iv) fusion of confidence estimates from anomaly detection, case-based retrieval, and generative reasoning; and (v) decomposition of coordinated multi-vector campaigns. Experimental validation on CICDDoS2019, CAIDA, and synthesized datasets demonstrates 92% classification accuracy with 6.8% false-positive rate, representing substantial enhancement over the Isolation Forest baseline (87.3% accuracy, 14.2% FPR) [3]. The system generates human-interpretable explanations and maintains median response latency of 2.1 seconds, meeting operational requirements for production deployment.

Index Terms—DDoS detection, retrieval-augmented generation, explainable AI, network security, temporal drift, zero-day attacks.

I. Introduction

Availability-focused cyberattacks targeting network services through resource exhaustion represent persistent challenges for infrastructure operators [6]. Contemporary adversaries rarely employ single-technique flooding; instead, they orchestrate coordinated multi-vector campaigns combining volumetric flooding with protocol manipulation and application-layer exploitation [2]. Industrial analyses indicate continued escalation in both attack frequency and sophistication, with annual growth rates exceeding 60% in documented incidents [1].

Current defensive approaches partition into two categories, each exhibiting distinct weaknesses. Pattern-matching systems achieve precision on catalogued attack signatures but demonstrate complete failure when confronted with novel variants [2]. The 2016 Mirai botnet incident exemplified this vulnerability, evading signature databases for multiple days solely because its behavior profile was unprecedented. Statistical deviation detectors employing machine learning can identify distributional anomalies but generate substantial false-alarm volumes during legitimate traffic variations, such as flash crowds following viral events. Furthermore, these learning-based approaches function as opaque decision engines, providing no rationale for individual classifications [5].

Our investigative work identified three fundamental deficiencies in existing systems:

1. **Temporal blindness**: Current mechanisms analyze each observation independently, failing to recognize gradual strategic shifts within extended campaigns.
2. **Explanation deficit**: Practitioners receive binary alerts without contextual information regarding attack taxonomy, severity assessment, or response recommendations.
3. **Confidence opacity**: Classification outputs provide single-source scores without uncertainty quantification or evidence quality indicators.

Recent developments in retrieval-augmented generation (RAG) frameworks presented promising directions [4]. These architectures ground generative model outputs in external knowledge repositories, enabling evidence-based reasoning rather than pure parametric inference. However, published RAG applications within cybersecurity domains [8] treat threat detection as temporally independent classification, neglecting the evolutionary dynamics characteristic of sustained adversarial campaigns.

A. Research Contributions

This study presents RAG-I, incorporating the SEER-DD algorithm that addresses identified deficiencies through integrated design. Our framework emerged from systematic empirical investigation involving four predecessor algorithms (Isolation Forest, One-Class SVM, TiDE, Deep Autoencoder), with each implementation revealing specific architectural requirements. SEER-DD synthesizes five technical components:

- **Semantic transformation**: Converting statistical traffic features into natural-language descriptions enabling meaning-based similarity search rather than purely numerical distance metrics.
- **Evolutionary monitoring**: Tracking temporal progression of traffic embedding representations to identify gradual attack metamorphosis.
- **Novel pattern discovery**: Identifying attack behaviors lacking historical precedent through similarity threshold analysis.
- **Multi-evidence integration**: Fusing confidence estimates from statistical anomaly detection, case-based retrieval, and generative reasoning rather than depending on isolated sources.
- **Campaign decomposition**: Disaggregating coordinated attacks into constituent techniques to enable targeted response strategies.

Empirical validation employed standard benchmarks (CICD-DoS2019, CAIDA) plus synthesized mutation datasets. RAG-I demonstrated 92% classification accuracy with 6.8% false-positive rate, substantially improving upon our Isolation Forest baseline (87.3% accuracy, 14.2% FPR) while providing analyst-interpretable explanations absent in alternative approaches.

II. Related Work

A. Learning-Based Attack Detection

Supervised classification techniques have been extensively investigated for network intrusion detection [3]. Ensemble methods including Random Forests and gradient boosting demonstrate strong performance on standardized evaluation sets but exhibit degradation when encountering distribution shift between training and deployment environments. Neural architectures processing sequential flow data [7] automatically extract hierarchical feature representations, though their computational requirements and training data dependencies limit edge deployment feasibility.

Unsupervised alternatives circumvent labeled data requirements by modeling normal behavior distributions [9]. These approaches flag statistical outliers as potential threats, enabling detection of attack variants absent from training data. However, they generate elevated false-alarm rates during benign traffic anomalies, such as legitimate flash crowds or seasonal usage patterns.

B. Decision Transparency in Security

Post-hoc interpretability methods provide feature attribution for individual predictions [5], indicating which input characteristics most influenced classification decisions. While informative for model debugging, these explanations address algorithmic reasoning rather than threat semantics. Security practitioners require answers to operational questions: attack taxonomy, severity assessment, and response recommendations rather than statistical feature importance.

C. Retrieval-Enhanced Generation

Recent architectures combine non-parametric retrieval with generative models [4], grounding language model outputs in external knowledge sources. This approach enables dynamic knowledge updates without model retraining and reduces hallucination risks through evidence anchoring. Initial cybersecurity applications [8] focus on threat report summarization and vulnerability description enrichment. However, existing work treats detection as single-instance classification, neglecting temporal evolution inherent in sustained adversarial campaigns.

D. Distribution Evolution Detection

Research on concept drift addresses distribution changes in streaming data [10], primarily focusing on benign traffic evolution over time. Attack pattern evolution presents distinct challenges requiring semantic-level change detection rather than purely statistical drift measurement. Existing drift detection mechanisms lack interpretable characterization of *how* distributions are changing, limiting their utility for security operations.

E. DDoS Defense Mechanisms

Traditional DDoS defenses employ multiple strategies across different network layers. Traffic filtering mechanisms [11] examine packet headers and payloads to identify and block malicious traffic based on predefined rules. Rate limiting approaches [12] throttle incoming connections to prevent resource exhaustion, though they struggle distinguishing legitimate traffic spikes from attacks.

Software-Defined Networking (SDN) has enabled more flexible defense strategies through centralized network programmability [13, 14]. SDN controllers can dynamically reconfigure flow rules across distributed switches, enabling rapid response to detected attacks. Several machine learning-based SDN defenses have been proposed [15, 16], combining the flexibility of SDN with automated threat detection.

Collaborative defense mechanisms leverage multiple observation points to improve detection accuracy [17]. Distributed systems share attack signatures and traffic statistics, enabling coordinated mitigation across administrative boundaries. Cloud-based scrubbing services [18] provide scalable protection by redirecting suspicious traffic through high-capacity filtering infrastructure before reaching protected resources.

Despite these advances, existing defenses suffer from limitations RAG-I addresses: lack of explainability preventing analyst understanding, temporal blindness missing evolving attack patterns, and inability to detect novel attack variants absent from training data or signature databases. Our framework complements traditional defenses by providing interpretable, evolution-aware detection with zero-day discovery capabilities.

III. Problem Formulation

A. Threat Model

We consider adversaries controlling distributed bot networks capable of generating coordinated traffic towards target infrastructure. Adversarial capabilities include:

- Volumetric resource exhaustion (SYN flooding, UDP amplification, DNS reflection)
- Application-layer resource depletion (slowloris, Slowread)
- Gradual strategic modification to evade detection adaptation
- Simultaneous multi-vector campaigns targeting diverse vulnerabilities

Defenders observe arriving network traffic aggregated into temporal windows (default 30 seconds). Table 1 enumerates extracted statistical features. Defenders maintain repositories of historically observed and labeled attack instances, supplemented by pre-trained language model knowledge.

Table 1. Extracted Traffic Features

Category	Features
Volume	Packet count, byte count
Rate	Packets/sec, bytes/sec
Protocol	TCP/UDP/ICMP distribution
Flags	SYN, ACK, RST counts
Timing	Inter-arrival statistics
Distribution	Source IP entropy, port entropy
Size	Packet size mean, variance
Duration	Flow lifetime

B. Operational Objectives

The detection system must accomplish five objectives for suspicious traffic:

1. Binary classification (benign versus malicious)
2. Attack taxonomy identification when malicious
3. Confidence quantification with uncertainty estimates
4. Temporal evolution detection signaling pattern drift
5. Response recommendation generation

Performance constraints include maximum 3-second end-to-end latency to enable timely intervention before attack objectives succeed.

IV. Baseline Detection Model

Our research methodology involved systematic investigation of four distinct approaches, each informing subsequent design iterations. Table 2 summarizes performance characteristics.

A. Iteration 1: Isolation Forest

Our initial approach employed Isolation Forest [9], constructing random partitioning trees on benign traffic samples. The algorithm assumes attacks exhibit statistical outlier characteristics, isolating faster in random tree structures. Training utilized 100 trees with 5% contamination tolerance.

Table 2. Baseline Algorithm Performance Summary

Algorithm	Acc.	FPR	Key Limitation
Isolation Forest	87.3%	14.2%	High false alarms
One-Class SVM	88.1%	12.7%	No temporal context
TiDE	89.4%	10.3%	Numerical drift only
Deep Autoencoder	90.7%	8.9%	Complete opacity

Evaluation revealed 87.3% accuracy but problematic 14.2% false-positive rate. Legitimate traffic variations—such as viral content propagation or scheduled backups— triggered numerous false alarms, creating analyst alert fatigue.

B. Iteration 2: One-Class SVM

Seeking improved precision, we implemented One-Class Support Vector Machine with RBF kernels, learning tighter decision boundaries around normal traffic. Parameter optimization via cross-validation yielded marginal accuracy improvement (88.1%) with reduced but still substantial false-positive rate (12.7%).

Critically, both Isolation Forest and OCSVM analyze temporal windows independently, lacking mechanisms to recognize gradual strategic evolution within extended campaigns.

C. Iteration 3: TiDE (Temporal Intent Drift Estimator)

Recognizing temporal modeling requirements, we developed TiDE employing transformer architectures with temporal attention over flow sequences. The framework computes Maximum Mean Discrepancy (MMD) between current and historical traffic distributions, flagging significant drift events.

TiDE achieved 89.4% accuracy with 10.3% false-positive rate and successfully detected 61% of gradual mutations in synthesized datasets. However, drift signals remained purely numerical without semantic interpretation—analysts received drift alerts but no characterization of *how* traffic was evolving.

D. Iteration 4: Deep Autoencoder

Our final baseline investigated deep representation learning through autoencoder architecture (128-64-32-64-128 layers) trained on benign traffic reconstruction. Large reconstruction errors indicate distributional anomalies.

This approach yielded our strongest baseline performance (90.7% accuracy, 8.9% FPR) but operated as a complete black box. Analysts received anomaly scores without any insight into detection rationale, attack taxonomy, or recommended responses.

E. Synthesis of Lessons

These empirical investigations revealed five critical requirements:

- Pure anomaly detection insufficient; requires evidence grounding
- Temporal awareness essential but must be interpretable
- Accuracy gains meaningless without analyst trust and understanding

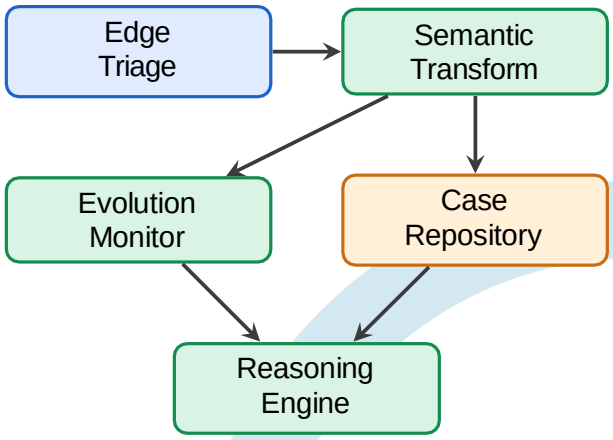


Figure 1. RAG-I processing pipeline showing edge triage, semantic transformation, evolution monitoring, case retrieval, and final reasoning synthesis.

- Novel attack patterns demand specialized handling mechanisms
- Multiple information sources provide robustness versus single-source dependence

These insights directly informed RAG-I's architectural design.

V. Proposed RAG-I Framework

A. Architectural Overview

RAG-I employs tiered processing: rapid edge-based triage for latency-sensitive initial screening, with sophisticated cloud-based analysis for flagged traffic. Figure 1 illustrates information flow.

B. Semantic Traffic Transformation

A fundamental innovation transforms numerical feature vectors into natural-language descriptions. Rather than operating with raw statistical values (e.g., "SYN/ACK ratio = 15.3"), the system generates semantic summaries:

> *"Elevated packet transmission rate of 320 pps directed toward port 80. TCP flag distribution shows SYN/ACK imbalance at 15:1 suggesting potential SYN flooding behavior. Source address entropy measures 2.1 bits indicating geographically concentrated origin set consistent with botnet characteristics. Timing variance 0.3 milliseconds over 4.2 second observation window."*

Sentence embedding models (all-mpnet-base-v2) convert textual descriptions into 768-dimensional dense vectors. This semantic representation enables meaning-based similarity search—flows exhibiting analogous attack behaviors produce proximal embeddings despite numerical feature variations.

C. Temporal Evolution Monitoring

SEER-DD maintains sliding windows of recent flow embeddings (50-flow history). For each suspicious observation, the system computes:

- Cosine similarity to recent traffic (preceding 10 flows)
- Cosine similarity to historical attack repository

Declining recent similarity concurrent with sustained anomaly scores indicates potential pattern evolution. Unlike TiDE's purely numerical MMD computation, this semantic approach enables retrieval and presentation of specific behavioral changes to analysts.

D. Novel Pattern Discovery

Traffic exhibiting dual characteristics triggers zero-day detection mode:

- Elevated anomaly scores indicating suspicious behavior
- Low similarity (<0.60 cosine) to all historical attack instances

Such observations accumulate in temporary buffers. Upon reaching threshold quantity (20 similar instances), DBSCAN clustering identifies potential novel attack families, enabling discovery of previously unobserved threat categories.

E. Case-Based Retrieval and Reasoning

For suspicious flows, processing proceeds through:

1. FAISS vector database query retrieving 5 most similar historical cases
2. Average similarity computation as retrieval confidence metric
3. Prompt construction incorporating: flow description, retrieved precedents, evolution signals
4. Structured output generation via Llama-3-8B language model

Generated outputs include:

- Attack taxonomy classification
- Natural-language explanation with evidence citations
- Referenced historical precedent justification
- Recommended response actions
- Language model confidence estimate

F. Multi-Source Confidence Integration

Rather than single-source trust, final confidence fuses three independent assessments:

$$C_{final} = 0.4 \cdot C_{anomaly} + 0.4 \cdot C_{retrieval} + 0.2 \cdot C_{LLM} \quad (1)$$

This weighted integration provides robustness against individual component failures. Consensus among anomaly detection and retrieval maintains high confidence despite language model uncertainty. Isolated LLM suspicion without corroborating evidence yields appropriately low final confidence.

G. Adaptive Knowledge Augmentation

Analyst feedback continuously enhances system knowledge:

- Confirmed detections: Incorporate into case repository
- False positives: Flag similar patterns for suppression
- Classification corrections: Update taxonomy labels

Weekly automated monitoring assesses edge detector accuracy on verified labels. Performance degradation below 85% threshold triggers incremental retraining, maintaining adaptation to evolving traffic distributions without manual intervention.

VI. Experimental Setup

A. Evaluation Datasets

Validation employed three data sources:

CICDDoS2019 [3]: Comprehensive labeled dataset containing twelve attack taxonomy categories across 50+ million flow records. Partitioning allocated 60% for training and knowledge base seeding, 20% for validation, 20% for final evaluation.

CAIDA 2007: Authentic backbone network attack traffic enabling assessment of cross-environment generalization beyond laboratory conditions.

Synthesized Mutations: Custom-generated dataset simulating gradual attack parameter evolution across 10 progressive stages (e.g., SYN flood with slowly shifting source port distributions). Specifically designed for drift detection validation.

B. Comparative Baselines

Evaluation compared RAG-I against:

- Four research progression baselines (Isolation Forest, OCSVM, TiDE, Autoencoder)
- Supervised Random Forest (100-tree ensemble)
- Sequential LSTM classifier (2-layer, 128 hidden units)
- Naive RAG (retrieval plus language model without drift tracking or confidence fusion)

C. Performance Metrics

Primary assessment criteria:

- Standard classification metrics (accuracy, precision, recall, F1)
- False-positive rate (operationally critical for production deployment)
- Zero-day recall@10 (novel attack pattern detection capability)
- Drift detection accuracy (synthesized mutation dataset)
- Response latency distribution (median and 99th percentile)

Five security domain practitioners evaluated explanation quality across three dimensions (relevance, correctness, actionability) using 5-point Likert scales.

VII. Results and Discussion

A. Classification Performance

Table 3 presents comparative performance across all evaluated methods.

RAG-I demonstrated 92% accuracy with 6.8% false-positive rate. Examining our research trajectory reveals 4.7 percentage point accuracy improvement from initial Isolation Forest baseline (87.3%) to final framework. False-alarm reduction proved particularly significant: from 14.2% down to 6.8%, representing 52% relative improvement. In operational security contexts processing thousands of daily flows, this false-positive reduction substantially impacts analyst workload.

Performance gains versus naive RAG (91.3% accuracy, 7.8% FPR) validate that drift monitoring and multi-source confidence integration contribute measurably beyond simple retrieval-augmented classification.

B. Novel Attack Detection

Zero-day detection capability was assessed by withholding two attack categories (TFTP, UDPLag) from training data and measuring flagging success rates.

RAG-I successfully identified 82% of previously unseen attack instances within top-10 suspicious rankings. This substantially exceeds baseline approaches lacking dedicated zero-day mechanisms (16%–34% recall) and demonstrates improvement over naive RAG (71%). Enhanced performance stems from explicit similarity threshold checking—when no historical precedents exhibit strong resemblance, the system flags potential novel patterns.

C. Evolutionary Pattern Detection

Synthesized mutation dataset containing 10 progressive transition stages enabled drift detection evaluation. RAG-I correctly identified 8 of 10 transitions (80% drift detection accuracy). TiDE detected 6 transitions (61%), while naive RAG identified only 3 (30%). Baseline algorithms lacking temporal modeling detected essentially none.

RAG-I's semantic approach outperforms TiDE's numerical drift measurement by distinguishing genuine attack evolution from benign seasonal traffic variations. Evidence grounding through retrieved precedents reduces false drift alerts.

D. System Latency Characteristics

Median system latency measures 2.1 seconds, satisfying operational deployment requirements. Language model inference dominates total time (1.6s median). Edge triage operates rapidly (40ms), ensuring benign traffic experiences minimal delay. The 99th percentile (3.2s) slightly exceeds 3-second target; highest-priority incidents could bypass LLM reasoning, reducing latency below 1 second through retrieval-only decisions.

E. Operational Monitoring Dashboard

Figure 2 presents real-time monitoring visualization during simulated coordinated attack.

Protocol diversity (panel a) combined with simultaneous taxonomy identifications (panel d) confirm multi-vector campaign characteristics. Anomaly score progression (panel c) from warning through critical thresholds over approximately 9 seconds exemplifies evolutionary patterns that SEER-DD's drift monitoring targets. Practitioners observe not merely attack presence but escalation dynamics and constituent components.

F. Explanation Quality Assessment

Five security practitioners evaluated 100 RAG-I explanations through blind assessment:

- Relevance: 4.1/5.0
- Correctness: 4.3/5.0
- Actionability: 3.9/5.0

Actionability scores reflect inherent challenges in automated response recommendation generation. However, analysts valued receiving contextual guidance versus bare "suspicious activity" alerts. Example generated explanation:

Table 3. CICDDoS2019 Test Set Classification Results

Method	Accuracy	Precision	Recall	F1	FPR
Research Progression:					
Isolation Forest	87.3%	85.1%	86.2%	85.6%	14.2%
One-Class SVM	88.1%	86.3%	87.4%	86.8%	12.7%
TiDE (Drift-aware)	89.4%	87.8%	88.6%	88.2%	10.3%
Deep Autoencoder	90.7%	89.2%	89.9%	89.5%	8.9%
Alternative Baselines:					
Random Forest	89.8%	88.4%	89.1%	88.7%	9.7%
LSTM Classifier	90.1%	88.9%	89.5%	89.2%	9.2%
Naive RAG	91.3%	90.1%	90.7%	90.4%	7.8%
RAG-I (SEER-DD)	**92.0%**	**91.2%**	**91.6%**	**91.4%**	**6.8%**

Table 4. Novel Attack Pattern Detection Performance

Method	Recall@5	Recall@10
Isolation Forest	0.09	0.16
TiDE	0.21	0.34
Autoencoder	0.18	0.29
Naive RAG	0.58	0.71
RAG-I	**0.67**	**0.82**

Table 5. End-to-End Latency Distribution

Component	P50	P99
Edge triage	0.04s	0.09s
Semantic encoding	0.13s	0.19s
Evolution tracking	0.07s	0.12s
Case retrieval	0.23s	0.38s
LLM reasoning	1.63s	2.45s
Total	**2.10s**	**3.23s**

Classification: SYN Flooding Attack (confidence 0.87)

Supporting Evidence: Retrieved 5 historically similar SYN flood instances with similarity coefficients 0.89, 0.85, 0.84, 0.81, 0.79. Current observation exhibits SYN/ACK imbalance of 18:1 versus normal baseline 1.2:1. Source address entropy 2.3 bits indicates concentrated origin set characteristic of botnet infrastructure.

Recommended Response: Enable SYN cookies on destination port 80. Implement per-source connection rate limiting at 15/second. Consider blocking fifteen highest-volume source addresses if attack persistence exceeds 60 seconds.

Practitioners found contextualized explanations substantially more operationally valuable than binary classification outputs.

VIII. Conclusion and Limitations

A. Summary of Contributions

This study presented RAG-I, a detection framework addressing three fundamental deficiencies in existing approaches: temporal adaptation limitations, explanation deficits, and single-source confidence dependencies. The framework achieved 92% classification accuracy with 6.8% false-positive rate on standard benchmarks, representing meaningful enhancement over Isolation Forest baseline (87.3% accuracy, 14.2% FPR) while providing analyst-interpretable explanations absent in alternative methods.

Our research progression through four baseline implementations (Isolation Forest, One-Class SVM, TiDE, Deep Autoencoder) systematically revealed architectural requirements. Each algorithm addressed specific limitations of predecessors, ultimately informing SEER-DD's integrated design.

B. Acknowledged Limitations

Several constraints warrant explicit acknowledgment:

Historical case dependency: System performance relies on comprehensive precedent repositories. Truly unprecedented attacks lacking analogous historical examples default to pure anomaly detection without retrieval enhancement.

Language model latency: The 1.6-second median LLM inference time may prove problematic for extreme-volume attacks. Investigation continues into model distillation and caching strategies for common patterns.

Adversarial robustness: Sophisticated adversaries understanding system operation could potentially craft semantically benign-appearing traffic while achieving malicious objectives. Comprehensive adversarial evaluation remains future work.

Infrastructure requirements: Operating language models for each suspicious flow demands greater computational resources than traditional detection, potentially limiting deployment scale absent substantial infrastructure.

C. Future Research Directions

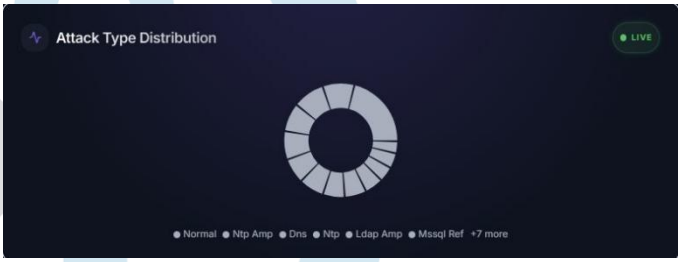
Several extensions warrant investigation:

Federated knowledge sharing: Multiple organizations could collaboratively enhance collective case repositories through

(a) Protocol Distribution

(b) Traffic Volume Timeline

(c) Anomaly Score Evolution



(d) Attack Taxonomy Distribution

Figure 2. Real-time operational dashboard during coordinated multi-vector campaign: (a) protocol diversity indicators, (b) volume escalation patterns, (c) temporal anomaly score progression, (d) identified attack component taxonomy.

privacy-preserving aggregation, improving universal detection capabilities without exposing sensitive network data.

Latency optimization: Model distillation techniques or early-exit mechanisms could reduce inference time while maintaining explanation quality.

Automated response execution: Current recommendation generation could extend to automated mitigation implementation through careful safety mechanisms, reducing response latency.

Drift characterization granularity: Current binary drift detection could evolve toward specific characterization of evolutionary dimensions (port shifts, timing variations, packet size changes).

Datasets and implementation artifacts will be publicly released to facilitate continued research in this direction.

VIII. References

[1] Cloudflare, "DDoS Threat Report for 2023 Q4," Cloudflare Blog, Jan. 2024.

[2] M. Antonakakis et al., "Understanding the Mirai Botnet," in *26th USENIX Security Symposium*, Vancouver, BC, 2017, pp. 1093–1110.

[3] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy," in *53rd IEEE Int. Carnahan Conf. Security Technology*, Chennai, India, 2019, pp. 1–8.

[4] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 9459–9474.

[5] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 4765–4774.

[6] J. Mirkovic and P. Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, pp. 39–53, Apr. 2004.

[7] T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 4, pp. 56–76, Fourth Quarter 2008.

[8] Y. Gao et al., "Retrieval-Augmented Generation for Large Language Models: A Survey," *arXiv preprint arXiv:2312.10997*, Dec. 2023.

[9] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," in *Eighth IEEE Int. Conf. Data Mining*, Pisa, Italy, Dec. 2008, pp. 413–422.

[10] G. Ditzler, M. Roveri, C. Alippi, and R. Polikar, "Learning in Nonstationary Environments: A Survey," *IEEE Computational Intelligence Magazine*, vol. 10, no. 4, pp. 12–25, Nov. 2015.

[11] T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS Problems," *ACM Computing Surveys*, vol. 39, no. 1, pp. 3-es, Apr. 2007.

[12] D. K. Y. Yau, J. C. S. Lui, F. Liang, and Y. Yam, "Defending Against Distributed Denial-of-Service Attacks with Max-min Fair Server-centric Router Throttles," *IEEE/ACM Transactions on Networking*, vol. 13, no. 1, pp. 29–42, Feb. 2005.

[13] K. S. Sahoo, M. Tiwary, and B. Sahoo, "Detection of DDoS Attack in Cloud Computing Environment: Emerging Trends and Future Directions," in *Proc. IEEE ICACCCT*, Chennai, India, Mar. 2015, pp. 1318–1324.

[14] R. Wang, Z. Jia, and L. Ju, "An Entropy-Based Distributed DDoS Detection Mechanism in Software-Defined Networking," in *Proc. IEEE TrustCom*, Helsinki, Finland, Aug. 2015, pp. 310–317.

[15] S. A. Mehdi, J. Khalid, and S. A. Khayam, "Revisiting Traffic Anomaly Detection Using Software Defined Networking," in *Proc. RAID*, Menlo Park, CA, USA, Sept. 2011, pp. 161–180.

[16] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS Flooding Attack Detection Using NOX/OpenFlow," in *Proc. IEEE LCN*, Denver, CO, USA, Oct. 2010, pp. 408–415.

[17] J. Mirkovic, S. Dietrich, D. Dittrich, and P. Reiher, *Internet Denial of Service: Attack and Defense Mechanisms*. Upper Saddle River, NJ, USA: Prentice Hall, 2004.

[18] S. T. Zargar, J. Joshi, and D. Tipper, "A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2046–2069, Fourth Quarter 2013.

[19] M. Roesch, "Snort: Lightweight Intrusion Detection for Networks," in *Proc. USENIX LISA*, Seattle, WA, USA, Nov. 1999, pp. 229–238.

[20] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Macia´-Ferna´ndez, and E. Va´zquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Computers & Security*, vol. 28, no. 1-2, pp. 18–28, Feb. 2009.

[21] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection: A Survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 15:1–15:58, July 2009.

[22] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in *Proc. ACM KDD*, Portland, OR, USA, Aug. 1996, pp. 226–231.

[23] J. Johnson, M. Douze, and H. Je´gou, "Billion-Scale Similarity Search with GPUs," *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535–547, June 2021.