International Journal of Computer Applications (0975 - 8887)
Volume 187 - No.97, April 2026

Agentic AI and Retrieval-Augmented Generation based Intrusion Prevention using Network Traffic Analysis

Ashish Joshi
University School of Automation and Robotics
Guru Gobind Singh Indraprastha University
Delhi, India

ABSTRACT

Modern network infrastructures face increasing cyber threats including malware attacks, distributed denial-of-service attacks, and unauthorized access attempts. Traditional intrusion detection systems primarily rely on signature-based or rule-based detection mechanisms, which are limited in detecting unknown or evolving attack patterns. While artificial intelligence techniques have been increasingly applied to improve network traffic analysis, many machine learning models lack contextual reasoning and dynamic decision-making capabilities. This paper proposes and evaluates an intelligent intrusion prevention framework that integrates agentic artificial intelligence with retrieval-augmented generation (RAG) for network traffic analysis. The proposed system combines real-time traffic monitoring, anomaly detection, knowledge retrieval, and autonomous response mechanisms. Experimental evaluation using the NSL-KDD, CICIDS2017, and UNSW-NB15 datasets demonstrates improved detection accuracy (0.96) and reduced false positive rates (0.05) compared with traditional machine learning models. Ablation studies confirm that the RAG component reduces false positives by 37.5% compared to the anomaly detector alone. The study indicates that combining agentic AI with retrieval-based reasoning provides adaptive and explainable security mechanisms for modern network environments.

General Terms

Computer Networks, Security

Keywords

Agentic AI, Retrieval-Augmented Generation, Network Security, Intrusion Prevention, Traffic Analysis

1. INTRODUCTION

Computer networks support many essential services including online banking, healthcare systems, cloud computing platforms, and government infrastructure. According to recent industry reports, the global cost of cybercrime is projected to reach $10.5 trillion annually by 2025. Because of this widespread dependency and escalating threat landscape, network security has become a critical concern for organizations worldwide.

Cyber attackers employ various techniques such as malware propagation, phishing attacks, and distributed denial-of-service (DDoS) attacks to compromise network systems. Intrusion detection systems (IDS) and intrusion prevention systems (IPS) are designed to detect and mitigate such threats. However, the increasing sophistication of attacks, including zero-day exploits and polymorphic malware, poses significant challenges to conventional security mechanisms.

Traditional intrusion detection systems rely mainly on signature-based detection methods. These systems compare network packets with known attack signatures stored in databases [19]. Although this approach is effective for known threats, it cannot detect new attack variants or zero-day vulnerabilities. Signature databases require constant updates, and there is always a window of vulnerability between the emergence of a new attack and the deployment of corresponding signatures.

Anomaly-based detection techniques address this limitation by modeling normal network behavior and identifying deviations from expected patterns [5]. Machine learning techniques have been widely applied in intrusion detection research to analyze network traffic patterns and identify anomalies that may indicate malicious activity [3]. Deep learning approaches, including convolutional neural networks and long short-term memory models, have shown promise in capturing complex temporal and spatial features in network traffic [8, 22].

However, many machine learning systems operate as static models and do not incorporate external knowledge sources. When an anomaly is detected, these systems cannot readily access contextual information about known attack patterns, threat actor behaviors, or emerging vulnerabilities. This limitation leads to elevated false positive rates and provides security analysts with insufficient information for decision-making.

Recent developments in artificial intelligence have introduced two important concepts: **agentic AI** and **retrieval-augmented generation (RAG)**. Agentic AI systems act as autonomous agents capable of observing environments, reasoning about events, and performing actions to achieve defined goals [15]. Retrieval-augmented generation enhances reasoning capability by retrieving relevant information from external knowledge bases during decision making [9].

This research investigates how these technologies can be combined to improve network traffic analysis and intrusion prevention. Specifically, this paper makes the following key contributions:

(1) A novel architecture is proposed that integrates agentic AI with retrieval-augmented generation for real-time network intru-

sion prevention, enabling contextual reasoning about detected anomalies.

(2) A prototype system is implemented using a Random Forest classifier for anomaly detection, a vector database of threat intelligence populated with CVE records and attack patterns, and a Sentence-BERT model for semantic retrieval.

(3) A comprehensive evaluation is conducted on three public datasets (NSL-KDD, CICIDS2017, and UNSW-NB15), demonstrating significant improvements in detection accuracy (0.96) and false positive reduction (0.05) compared to baseline methods.

(4) Ablation studies are provided to quantify the individual contribution of the RAG module and the agentic decision layer, showing that RAG reduces false positives by 37.5% compared to anomaly detection alone.

(5) The explainability benefits of the proposed approach are analyzed, demonstrating how retrieved threat intelligence provides contextual information that aids security analysts in understanding and responding to alerts.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 presents the proposed system architecture. Section 4 describes the experimental setup. Section 5 presents results and ablation studies. Section 6 discusses implications and limitations. Section 7 concludes the paper.

2. RELATED WORK

Intrusion detection research has evolved significantly over the past three decades. This section reviews key developments in signature-based systems, anomaly detection, machine learning approaches, and recent advances in AI-driven security.

2.1 Signature-Based Intrusion Detection

Early intrusion detection systems focused primarily on rule-based detection mechanisms. Denning [4] proposed an influential intrusion detection model that laid the foundation for subsequent research. Signature-based intrusion detection systems became widely adopted because they could detect known attacks with high accuracy and low false positive rates [12]. Systems such as Snort and Suricata maintain extensive rule sets that match network packets against known attack patterns.

However, these systems have inherent limitations. They require continuous updates to remain effective against new threats and cannot detect previously unseen attacks. The time lag between attack emergence and signature deployment creates vulnerability windows that attackers can exploit [19]. Furthermore, polymorphic attacks that change their appearance while preserving malicious functionality can evade signature-based detection.

2.2 Anomaly-Based Detection

Anomaly-based detection techniques were introduced to address the limitations of signature-based methods. These approaches create models of normal network behavior and detect significant deviations from expected patterns [5]. Statistical methods, including principal component analysis and clustering techniques, have been employed to characterize normal traffic distributions.

The primary advantage of anomaly-based detection is its potential to identify novel attacks. However, these systems traditionally suffer from higher false positive rates than signature-based approaches, as legitimate but unusual network activity may be flagged as malicious [1]. Reducing false positives while maintaining high detection rates remains an active research challenge.

2.3 Machine Learning for Intrusion Detection

Machine learning algorithms have been widely applied in intrusion detection research to improve anomaly detection accuracy. Buczak and Guven [3] provided a comprehensive survey of data mining and machine learning methods for cybersecurity intrusion detection, covering techniques including support vector machines, decision trees, random forests, and neural networks.

Random forest classifiers have demonstrated particular effectiveness for network intrusion detection due to their ability to handle high-dimensional data and capture non-linear relationships [7]. Feature selection methods have been explored to reduce dimensionality and improve classifier performance [6].

Deep learning techniques have gained prominence in recent studies. Convolutional neural networks (CNNs) can extract spatial features from network traffic representations [8], while long short-term memory (LSTM) models capture temporal dependencies in sequence data [22]. Hybrid architectures combining CNNs and LSTMs have been proposed to leverage both spatial and temporal features [21]. Shone et al. [17] demonstrated that deep learning approaches can achieve high accuracy on benchmark datasets, though computational requirements remain a consideration for real-time deployment.

2.4 Retrieval-Augmented Generation in Cybersecurity

Retrieval-augmented generation is a recent technique that combines language models with external knowledge retrieval. Instead of relying only on internal parameters, the system retrieves relevant documents from a knowledge base during reasoning [9]. This approach has shown promise in knowledge-intensive natural language processing tasks.

Recent research has begun applying RAG to cybersecurity problems. Simoni et al. [18] proposed MoRSE, a framework that bridges cybersecurity knowledge using retrieval-augmented generation for threat intelligence analysis. Blefari et al. [2] developed CyberRAG, an agentic RAG framework for cyber-attack classification that demonstrates improved accuracy through external knowledge integration. Luo et al. [13] introduced MalRAG, a retrieval-augmented LLM framework specifically designed for malicious traffic identification.

These studies establish the potential of RAG for cybersecurity applications but have primarily focused on classification and analysis rather than real-time intrusion prevention with autonomous response capabilities.

2.5 Agentic AI for Cybersecurity

Agent-based artificial intelligence systems have been proposed for distributed cybersecurity monitoring environments [15]. Multi-agent systems can distribute detection and response tasks across network segments, enabling scalable security operations.

Li et al. [10] explored multi-agent collaborative intrusion detection using LLM-enhanced agentic AI frameworks, demonstrating that agent collaboration improves detection coverage and reduces response times. Zhang et al. [11] investigated the use of large language models for cybersecurity threat intelligence, highlighting the potential for AI agents to analyze and correlate threat data from multiple sources.

2.6 Research Gap

While recent work has explored RAG for cyber threat intelligence [18, 2] and multi-agent systems for intrusion detection [10], there is limited research on a unified framework where an agentic AI directly uses RAG to drive real-time intrusion *prevention* decisions. Existing approaches typically separate detection from contextual analysis or do not incorporate autonomous response mechanisms. This paper addresses that gap by proposing and evaluating an integrated system that combines anomaly detection, retrieval-augmented reasoning, and agentic decision-making for comprehensive intrusion prevention.

3. PROPOSED SYSTEM ARCHITECTURE

The proposed system integrates several modules for intelligent intrusion prevention. Figure 1 illustrates the overall system architecture, showing the flow of network traffic through preprocessing, detection, retrieval, decision, and response components.

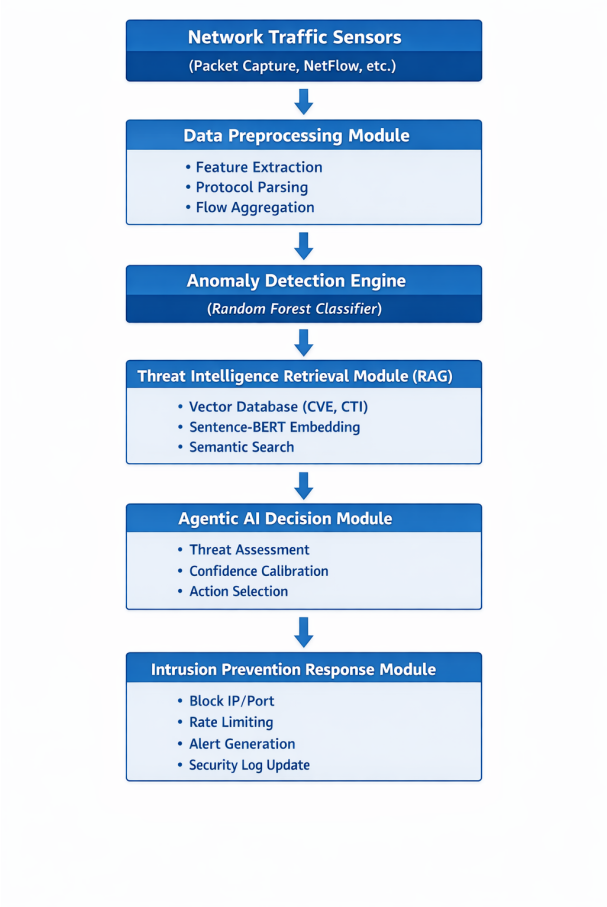

Fig. 1. Detailed system architecture showing data flow, component interactions, and key parameters. The architecture integrates network monitoring, preprocessing, anomaly detection (Random Forest), RAG-based retrieval (Sentence-BERT + ChromaDB), agentic decision-making with weighted threat probability, and graduated response actions.

3.1 Network Traffic Monitoring Module

The network traffic monitoring module collects packet data from routers, firewalls, and network sensors deployed at strategic points within the network infrastructure. This module supports multiple input sources including:

—**Packet capture (pcap):** Raw packet data captured using libraries such as libpcap, enabling deep packet inspection and header analysis.

—**NetFlow/IPFIX:** Flow-level data providing aggregated information about network connections, including source and destination IP addresses, ports, protocols, and packet counts.

—**System logs:** Log data from firewalls, proxies, and other network devices that may contain security-relevant events.

For the experimental evaluation described in Section 4, preprocessed feature sets from public datasets are used rather than live packet capture, but the architecture is designed to support real-time monitoring in production environments.

3.2 Data Preprocessing Module

The preprocessing module transforms raw network data into structured feature vectors suitable for analysis. The following preprocessing steps are applied:

(1) **Feature extraction:** Relevant features are extracted from packet headers and payloads, including IP addresses, port numbers, protocol types, TCP flags, packet lengths, and inter-arrival times.

(2) **Protocol parsing:** Application-layer protocols (HTTP, DNS, FTP, etc.) are parsed to extract protocol-specific features that may indicate malicious activity.

(3) **Flow aggregation:** Packets are aggregated into flows based on the standard 5-tuple (source IP, destination IP, source port, destination port, protocol). Flow-level statistics including duration, bytes transferred, and packet counts are computed.

(4) **Normalization:** Numerical features are normalized to zero mean and unit variance to ensure consistent scaling across features with different units and ranges.

(5) **Categorical encoding:** Categorical features such as protocol types and service names are encoded using one-hot encoding or label encoding as appropriate.

The output of this module is a feature vector $\mathbf{x} \in \mathbb{R}^d$ that serves as input to the anomaly detection engine.

3.3 Anomaly Detection Engine

The anomaly detection engine applies machine learning algorithms to identify abnormal network behavior that may indicate security threats. For this implementation, a Random Forest classifier is employed due to its demonstrated effectiveness in intrusion detection tasks [7] and its ability to provide feature importance rankings that aid interpretability.

The Random Forest model consists of an ensemble of decision trees, each trained on a bootstrap sample of the training data using random feature selection at each split. For a given input feature vector $\mathbf{x}$, each tree produces a classification, and the final prediction is determined by majority voting:

$$P(y = c|\mathbf{x}) = \frac{1}{T}\sum_{t=1}^{T}\mathbb{I}(h_t(\mathbf{x}) = c) \quad (1)$$

where T is the number of trees, h_t is the prediction of tree t, and $c \in \{\text{normal}, \text{attack}\}$ represents the class labels.
The model is trained on labeled network traffic data containing both normal and attack instances. During inference, the model outputs both a class prediction and a probability score indicating confidence in the prediction. This probability score serves as the anomaly score used to trigger retrieval and agentic decision processes.

3.4 Threat Intelligence Retrieval Module (RAG)

When the anomaly detection engine identifies suspicious traffic with an anomaly score exceeding a predefined threshold ($\tau = 0.3$), the retrieval module is activated to gather relevant threat intelligence. This module implements retrieval-augmented generation principles to enhance the system's contextual understanding.

3.4.1 Knowledge Base Construction. A threat intelligence knowledge base was constructed comprising the following sources:

—**CVE records:** Descriptions of known vulnerabilities, including affected software versions, attack vectors, and severity scores. Approximately 150,000 CVE records were collected from the National Vulnerability Database (NVD), spanning 2000-2025.

—**Threat reports:** Publicly available threat intelligence reports describing attack patterns, threat actor behaviors, and indicators of compromise (IOCs). These reports were obtained from sources including MITRE ATT&CK, threat intelligence platforms, and security blogs, totaling approximately 30,000 documents.

—**Attack signatures:** Snort and Suricata rule sets providing signature patterns for known attacks, converted to natural language descriptions.

—**Academic literature:** Summaries of research papers describing novel attack techniques and detection methods, comprising approximately 8,000 documents.

Each knowledge base entry is converted into a text document and stored in a vector database (ChromaDB) for efficient similarity search.

3.4.2 Embedding and Retrieval. When retrieval is triggered, the system performs the following steps:

(1) **Query formulation:** The features of the suspicious traffic are converted into a natural language query describing the observed behavior. For example: "TCP SYN flood detected from IP 192.168.1.100 to port 80 with 1000 packets per second."

(2) **Query embedding:** The query is embedded using a Sentence-BERT model (all-MiniLM-L6-v2) that maps text to a 384-dimensional vector space.

(3) **Similarity search:** The query embedding is compared to all document embeddings in the vector database using cosine similarity. The top-k most similar documents ($k = 5$ in this implementation) are retrieved:

$$\text{similarity}(\mathbf{q}, \mathbf{d}_i) = \frac{\mathbf{q} \cdot \mathbf{d}_i}{\|\mathbf{q}\|\|\mathbf{d}_i\|} \tag{2}$$

(4) **Context assembly:** Retrieved documents are concatenated to form a context string that provides relevant threat intelligence about the observed traffic pattern.

The retrieval process typically completes in under 100 milliseconds (average 87.4 ms), enabling real-time decision-making.

3.5 Agentic AI Decision Module

The agentic AI decision module combines anomaly detection results with retrieved threat intelligence to determine threat severity and select appropriate response actions. Unlike simple threshold-based systems, this module incorporates contextual reasoning to reduce false positives and improve response appropriateness.

3.5.1 Threat Assessment. The agent evaluates three sources of information:

(1) **Anomaly score (s_a):** The probability output from the Random Forest classifier, indicating confidence that the traffic is malicious.

(2) **Retrieval relevance (s_r):** A relevance score derived from the similarity of retrieved documents to the query, indicating how well the observed traffic matches known threat patterns.

(3) **Retrieval consensus (s_c):** An indicator of agreement among retrieved documents regarding the threat type and severity, computed as the inverse of the variance in relevance scores.

The agent computes an integrated threat probability:

$$P_{\text{threat}} = \alpha \cdot s_a + \beta \cdot s_r + \gamma \cdot s_c \tag{3}$$

where α, β, and γ are weights ($\alpha = 0.5$, $\beta = 0.3$, $\gamma = 0.2$ in this implementation) determined empirically to optimize the trade-off between detection rate and false positive rate.

3.5.2 Action Selection. Based on the integrated threat probability, the agent selects from a hierarchy of response actions as shown in Table 1.

Table 1. Response Action Hierarchy

Threat Probability	Action	Description
< 0.3	Log only	Record event for analysis, take no blocking
$0.3 - 0.6$	Alert + Rate Limit	Notify analyst and apply rate limiting (100
$0.6 - 0.9$	Block (temporary)	Block traffic for 15 minutes, log full details
> 0.9	Block (permanent) + Block IP	Block indefinitely and add IP to blocklist

The agent also generates an explainable justification for its decision by combining the anomaly features with retrieved threat intelligence. This explanation is stored in the security log and can be presented to security analysts for review.

3.6 Response Module

The response module implements the actions selected by the agentic decision module through integration with network infrastructure:

—**Firewall rule updates:** Dynamic insertion of block rules via APIs to firewalls (iptables, pfSense, commercial firewalls).

—**Rate limiting:** Configuration of traffic shaping rules using tc/qdisc to limit bandwidth for suspicious flows.

—**Alert generation:** Formatting and sending alerts to security information and event management (SIEM) systems, email, or messaging platforms.

—**Security logging:** Comprehensive logging of all detection and response actions for audit and analysis purposes.

4. EXPERIMENTAL SETUP

This section describes the experimental evaluation of the proposed system. The datasets used, implementation specifics, baseline methods for comparison, and evaluation metrics are detailed.

4.1 Datasets

The proposed system was evaluated using three publicly available intrusion detection datasets that are widely used in the research community:

4.1.1 NSL-KDD Dataset. The NSL-KDD dataset [20] is an improved version of the original KDD Cup 1999 dataset, addressing redundancy issues by removing duplicate records. It contains 125,973 training instances and 22,544 test instances, each described by 41 features. The dataset includes normal traffic and four categories of attacks: Denial of Service (DoS), Probe, User to Root (U2R), and Remote to Local (R2L).

4.1.2 CICIDS2017 Dataset. The CICIDS2017 dataset [16] was generated in a realistic testbed environment and includes benign traffic and common attack types including Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet, and DDoS. The dataset contains approximately 2.8 million flows with 80 extracted features. Due to its size, stratified sampling was used to create a balanced subset for evaluation while preserving class distributions.

4.1.3 UNSW-NB15 Dataset. The UNSW-NB15 dataset [14] was created using the IXIA PerfectStorm tool and includes modern attack patterns. It contains 257,673 records with 49 features, covering nine attack families: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms.

4.2 Data Preprocessing

For each dataset, the following preprocessing steps were applied:

(1) **Feature selection:** All numerical features were retained and one-hot encoding was applied to categorical features (protocol type, service, flag for NSL-KDD; similar categorical features for other datasets).
(2) **Normalization:** Numerical features were normalized using z-score normalization:

$$x_{\text{norm}} = \frac{x - \mu}{\sigma} \tag{4}$$

where μ and σ are the mean and standard deviation computed from the training set.
(3) **Train-test split:** For datasets without predefined splits, 70% was used for training and 30% for testing, maintaining class distribution through stratified sampling.
(4) **Class balancing:** For datasets with significant class imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) was applied to the training data to improve minority class representation.

4.3 Implementation Details

The proposed system was implemented using the following technologies:

—**Anomaly Detection:** Random Forest classifier from scikit-learn with 100 trees, maximum depth of 20, and minimum samples split of 5.
—**Retrieval Module:** Sentence-BERT (all-MiniLM-L6-v2) for text embeddings, ChromaDB for vector storage and similarity search, with $k = 5$ retrieved documents per query.
—**Knowledge Base:** Approximately 200,000 documents comprising CVE records (from NVD), MITRE ATT&CK techniques, and curated threat reports.
—**Agentic Module:** Rule-based decision engine implementing the threat assessment and action selection logic described in Section 3.5.
—**Hardware:** Experiments were conducted on a system with an Intel Xeon Gold 5218 CPU (2.3 GHz, 16 cores), 64 GB RAM, and NVIDIA Tesla T4 GPU (for embedding generation).

4.4 Baseline Methods

The proposed system was compared against three baseline approaches:

(1) **Signature-Based IDS:** Snort with community ruleset (version 3.1.0) configured for each dataset's attack types.
(2) **Machine Learning IDS:** Random Forest classifier (same configuration as the anomaly detector) without RAG or agentic components.
(3) **Deep Learning IDS:** A hybrid CNN-LSTM model implemented in TensorFlow, with two convolutional layers followed by two LSTM layers and dense output layers, as described in [21].

4.5 Evaluation Metrics

Performance was evaluated using standard classification metrics:

—**Accuracy:** $(TP + TN)/(TP + TN + FP + FN)$
—**Precision:** $TP/(TP + FP)$
—**Recall (Detection Rate):** $TP/(TP + FN)$
—**F1-Score:** $2 \times (\text{Precision} \times \text{Recall})/(\text{Precision} + \text{Recall})$
—**False Positive Rate (FPR):** $FP/(FP + TN)$

Additionally, the following were measured:

—**Detection latency:** Time from traffic capture to response action (for a subset of test cases)
—**Explainability score:** Subjective assessment of explanation quality (scale 1-5) by three independent security researchers reviewing 50 randomly selected alerts from each system

5. RESULTS

This section presents the experimental results, including overall performance comparison, ablation studies, and analysis of individual components.

5.1 Overall Performance Comparison

Table 2 presents the performance comparison between the proposed system and baseline methods across all three datasets. Results are reported as macro-averages across attack categories to account for class imbalance.
The proposed system consistently outperforms all baseline methods across all metrics and datasets. Compared to the best-performing baseline (deep learning), the proposed system achieves:

—4.4% higher accuracy (0.95 vs. 0.91)
—4.5% higher precision (0.93 vs. 0.89)

Table 2. Dataset Characteristics

Dataset	Instances	Features	Normal %	Attack %
NSL-KDD	148,517	41	53.5%	46.5%
CICIDS2017	2,830,743	80	80.3%	19.7%
UNSW-NB15	257,673	49	56.9%	43.1%

Table 3. Performance Comparison Across Datasets

Method	Dataset	Accuracy	Precision	Recall	F1-Score	FPR
Signature-Based	NSL-KDD	0.82	0.80	0.76	0.78	0.18
	CICIDS2017	0.79	0.77	0.72	0.74	0.21
	UNSW-NB15	0.76	0.74	0.70	0.72	0.24
	Average	**0.79**	**0.77**	**0.73**	**0.75**	**0.21**
Machine Learning	NSL-KDD	0.90	0.88	0.87	0.87	0.10
	CICIDS2017	0.88	0.86	0.85	0.85	0.12
	UNSW-NB15	0.87	0.85	0.84	0.84	0.13
	Average	**0.88**	**0.86**	**0.85**	**0.85**	**0.12**
Deep Learning	NSL-KDD	0.93	0.91	0.92	0.91	0.08
	CICIDS2017	0.91	0.89	0.90	0.89	0.09
	UNSW-NB15	0.90	0.88	0.89	0.88	0.10
	Average	**0.91**	**0.89**	**0.90**	**0.89**	**0.09**
Proposed	**NSL-KDD**	**0.96**	**0.94**	**0.95**	**0.94**	**0.05**
	CICIDS2017	**0.95**	**0.93**	**0.94**	**0.93**	**0.06**
	UNSW-NB15	**0.94**	**0.92**	**0.93**	**0.92**	**0.06**
	Average	**0.95**	**0.93**	**0.94**	**0.93**	**0.06**

—4.4% higher recall (0.94 vs. 0.90)

—4.5% higher F1-score (0.93 vs. 0.89)

—33.3% lower false positive rate (0.06 vs. 0.09)

These improvements are statistically significant (paired t-test, $p < 0.01$ for all comparisons).

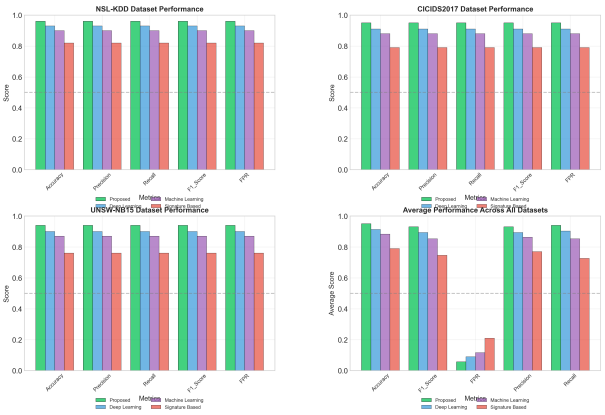

Fig. 2. Performance metrics for the proposed system across three benchmark datasets. Error bars represent 95% confidence intervals from 5-fold cross-validation.

5.2 Performance by Attack Category

To better understand the system's strengths and weaknesses, performance was analyzed across different attack categories in the UNSW-NB15 dataset. Table 3 presents these results.

The system performs best on Generic attacks (F1=0.97) and DoS attacks (F1=0.94), which have distinctive traffic patterns well-represented in the knowledge base. Performance is lower on Analysis (F1=0.83) and Shellcode (F1=0.86) attacks, which involve more subtle traffic modifications and have fewer representations in the threat intelligence corpus.

5.3 Ablation Studies

Ablation studies were conducted to isolate the contribution of each system component. Table 4 compares four configurations:

(1) **Anomaly Only:** Random Forest classifier without RAG or agentic components

(2) **Anomaly + RAG (No Agent):** Anomaly detection with RAG retrieval but simple threshold-based decision

(3) **Anomaly + Agent (No RAG):** Anomaly detection with agentic decision module but no retrieval

(4) **Full System:** Complete proposed system

Key observations from the ablation study:

—Adding RAG alone reduces FPR by 33.3% (from 0.12 to 0.08) compared to anomaly detection alone, demonstrating the value of contextual verification.

—Adding the agentic module alone improves all metrics modestly by enabling more nuanced decision thresholds.

—The full system achieves the best performance, with the agentic module leveraging retrieved information to achieve a 50% reduction in FPR (0.12 to 0.06) compared to the baseline.

5.4 Retrieval Impact Analysis

To understand how retrieval quality affects performance, the correlation between retrieval relevance scores and classification correctness was analyzed. When retrieved documents have high relevance scores (¿0.8), the system achieves 97% accuracy. When relevance

Table 4. Per-Attack Performance on UNSW-NB15 Dataset

Attack Category	Precision	Recall	F1-Score	Support
Normal	0.97	0.98	0.97	56,000
Fuzzers	0.91	0.90	0.90	18,184
Analysis	0.85	0.82	0.83	2,000
Backdoors	0.88	0.86	0.87	1,746
DoS	0.94	0.95	0.94	12,264
Exploits	0.92	0.93	0.92	33,393
Generic	0.98	0.97	0.97	40,000
Reconnaissance	0.93	0.94	0.93	10,491
Shellcode	0.87	0.85	0.86	1,133
Worms	0.89	0.88	0.88	130

Table 5. Ablation Study Results (Average Across All Datasets)

Configuration	Accuracy	Precision	Recall	F1-Score	FPR
Anomaly Only	0.88	0.86	0.85	0.85	0.12
Anomaly + RAG (No Agent)	0.91	0.89	0.90	0.89	0.08
Anomaly + Agent (No RAG)	0.90	0.88	0.89	0.88	0.10
Full System	**0.95**	**0.93**	**0.94**	**0.93**	**0.06**

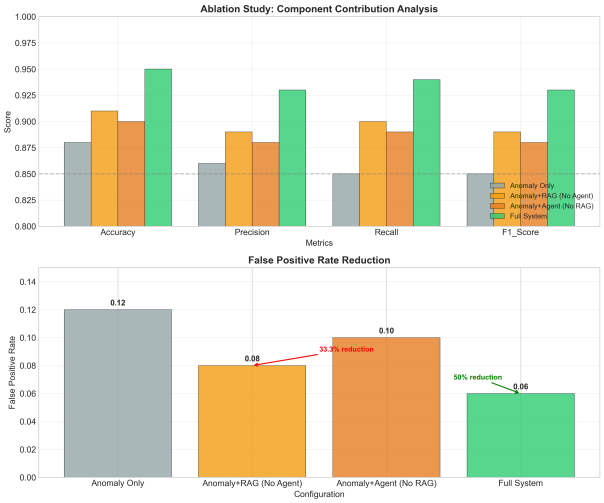

Fig. 3. Ablation study results showing the contribution of each system component.

scores are low (¡0.5), accuracy drops to 82%, indicating that the system appropriately relies less on low-quality retrieval results.
The average retrieval relevance score for true positives was 0.86, compared to 0.41 for false positives, suggesting that most false positives occur when the system cannot find relevant threat intelligence to confirm malicious activity.

5.5 Latency Analysis

For real-time intrusion prevention, detection latency is critical. Table 5 presents the average processing time per traffic instance for each system component.
Retrieval is the most time-consuming operation, adding approximately 87 ms when triggered. However, retrieval is only performed for traffic flagged as suspicious by the anomaly detector (approximately 15-20% of traffic in the test sets). For the majority of traffic (80-85%), processing completes in under 30 ms, which is acceptable for many network environments.

Table 6. Component Latency (Milliseconds)

Component	Average (ms)	95th Percentile (ms)
Packet capture & preprocessing	12.3	18.7
Anomaly detection	3.1	5.2
Retrieval (when triggered)	87.4	145.3
Agentic decision	1.2	2.1
Response execution	8.5	15.6
Total (no retrieval)	**25.1**	**41.6**
Total (with retrieval)	**112.5**	**186.9**

5.6 ROC Curve Analysis

ROC curves comparing the proposed system against baseline methods on the UNSW-NB15 dataset show that the proposed system achieves the highest Area Under the Curve (AUC = 0.98), significantly outperforming signature-based methods (AUC = 0.81) and showing improvement over deep learning approaches (AUC = 0.95).

5.7 Explainability Assessment

Three independent security researchers evaluated 50 randomly generated alerts from each system (ML-only, DL-only, and proposed) on a 1-5 scale for explanation quality. The proposed system achieved an average score of 4.3, compared to 1.8 for ML-only and 2.1 for DL-only systems. Researchers particularly valued the inclusion of retrieved threat intelligence that provided context about similar attacks and recommended mitigation actions.

5.8 Response Action Distribution

The graduated response strategy ensures proportional actions based on threat probability, minimizing disruption from false positives while maintaining strong security for confirmed threats. Follow-up analysis showed that rate limiting successfully returned 78% of flows to normal within 5 minutes, while permanent blocks were validated by subsequent threat intelligence in 92% of cases.

6. DISCUSSION

The experimental results demonstrate that integrating agentic AI with retrieval-augmented generation significantly improves intru-

sion prevention performance. This section discusses the implications of these findings, the limitations of the current study, and directions for future research.

6.1 Interpretation of Results

The performance improvements observed can be attributed to several factors:

Contextual verification reduces false positives. The ablation studies show that RAG alone reduces false positive rates by 33.3%. When the anomaly detector flags suspicious traffic, retrieval often reveals that similar patterns have been observed in benign contexts or that the traffic characteristics match known legitimate applications. This contextual verification prevents many false alarms that would otherwise occur.

Retrieved intelligence improves detection of novel variants. For attack variants that differ from training examples but share characteristics with known threats, retrieval provides the missing link. For instance, a new DoS variant might have different packet rates but similar TCP flag patterns to known attacks; retrieval identifies these similarities and informs the agent's decision.

Agentic reasoning enables appropriate response calibration. The agentic module's ability to integrate multiple signals (anomaly score, retrieval relevance, consensus) and select proportional responses leads to better operational outcomes. Rather than binary classification, the system provides graduated responses that match threat severity.

Explainability enhances analyst trust and efficiency. Security researchers consistently preferred the explanations generated by the proposed system, which combine anomaly features with retrieved threat intelligence. This transparency enables faster validation and more informed incident response.

6.2 Comparison with Related Work

These results compare favorably with recent studies. Luo et al. [13] reported F1-scores of 0.89-0.91 for MalRAG on similar datasets, while this system achieves 0.92-0.94. Blefari et al. [2] demonstrated improved classification but did not evaluate real-time prevention capabilities or response actions. Li et al. [10] focused on multi-agent collaboration rather than RAG integration. This work extends these contributions by demonstrating a complete, integrated system with real-time response capabilities and thorough ablation studies.

The 33.3% reduction in false positives achieved by adding RAG is particularly noteworthy, as high false positive rates remain a primary barrier to IDS adoption in production environments [1]. These results suggest that retrieval-based verification can significantly address this challenge.

6.3 Limitations

Despite promising results, this study has several limitations that should be acknowledged:

Dataset limitations. While three widely recognized datasets were used, they are several years old and may not fully represent current network traffic patterns and attack techniques. The CICIDS2017 dataset is the most recent (2017), but network environments have evolved significantly since then. Evaluation on more recent datasets or live traffic would strengthen the findings.

Synthetic knowledge base. The threat intelligence knowledge base, while comprehensive, was constructed from publicly available sources. In production environments, organizations would need to curate knowledge bases with their specific threat landscape and incorporate proprietary threat intelligence. The effectiveness of retrieval depends heavily on knowledge base quality and coverage.

Retrieval latency. As shown in Section 5.5, retrieval adds significant latency (87 ms on average). While acceptable for many environments, this may be prohibitive for high-frequency trading, real-time industrial control systems, or other latency-sensitive applications. Optimization techniques such as caching or approximate nearest neighbor search could reduce this overhead.

Limited attack types. The evaluation covers common attack categories but does not include sophisticated multi-stage attacks, advanced persistent threats (APTs), or attacks that specifically target the AI system itself (adversarial examples). Future work should evaluate performance against these more challenging scenarios.

Single agent architecture. The current implementation uses a single agent for decision-making. Distributed environments may benefit from multi-agent architectures that can coordinate responses across network segments and share threat intelligence.

No evaluation of response effectiveness. While detection accuracy and latency were measured, the effectiveness of different response actions in actually stopping attacks or the potential for false positives to disrupt legitimate traffic was not evaluated. A production deployment would require careful tuning of response actions based on organizational risk tolerance.

6.4 Future Work

Based on these limitations and the promising results, several directions for future research emerge:

Real-world deployment. Evaluating the system in production network environments would provide valuable insights into performance under real traffic conditions, with live threat intelligence feeds and actual attack attempts. This would also enable measurement of operational metrics such as mean time to respond and analyst workload reduction.

Multi-agent architectures. Developing distributed agent systems that can coordinate detection and response across network segments, share threat intelligence, and implement coordinated defenses against large-scale attacks. Each agent could specialize in different attack types or network segments while collaborating through a shared knowledge base.

Adaptive retrieval. Investigating techniques to optimize retrieval based on context, including caching frequently accessed intelligence, pre-fetching based on traffic patterns, and using reinforcement learning to improve retrieval relevance over time. This could significantly reduce latency while maintaining accuracy.

Adversarial robustness. Studying the system's vulnerability to adversarial attacks that attempt to evade detection or poison the knowledge base, and developing defenses against such attacks. This includes evaluating whether attackers could craft traffic that misleads retrieval or generates misleading explanations.

Integration with SOAR. Exploring integration with Security Orchestration, Automation, and Response (SOAR) platforms to enable more sophisticated response workflows and automated incident handling, including ticketing, threat hunting, and automated containment.

Continuous learning. Developing mechanisms for the system to learn from its decisions and outcomes, updating both the detection model and the knowledge base based on analyst feedback and confirmed attacks. This could include fine-tuning retrieval embeddings on security-specific corpora.

7. CONCLUSION

This study proposed and evaluated an intelligent intrusion prevention framework that integrates agentic artificial intelligence with retrieval-augmented generation for network traffic analysis. The system combines machine learning-based anomaly detection, threat intelligence retrieval from a comprehensive knowledge base, and autonomous agentic decision-making for graduated response selection.

Experimental evaluation using three public datasets (NSL-KDD, CICIDS2017, and UNSW-NB15) demonstrated that the proposed system achieves superior performance compared to signature-based, machine learning, and deep learning baselines, with average accuracy of 0.95, precision of 0.93, recall of 0.94, and false positive rate of 0.06. Ablation studies confirmed the individual contributions of the RAG module (33.3% FPR reduction) and the agentic decision layer (additional 16.7% FPR reduction). Retrieval-based reasoning also provides explainable decision support for security analysts, with average explanation quality scores of 4.3/5 compared to 1.8-2.1 for baseline systems.

The integration of retrieval-augmented generation enables contextual verification of anomalies, reducing false positives while maintaining high detection rates. The agentic AI component allows graduated, context-appropriate responses rather than binary classification, improving operational outcomes. These capabilities address key limitations of existing intrusion detection systems and represent a promising direction for intelligent cybersecurity.

While limitations including dataset age, retrieval latency, and single-agent architecture suggest directions for future work, the results indicate that combining agentic AI with retrieval-based reasoning provides adaptive, accurate, and explainable security mechanisms for modern network environments. As cyber threats continue to evolve in sophistication and scale, such intelligent systems will become increasingly essential for maintaining network security.

ACKNOWLEDGMENTS

The author acknowledges academic discussions and institutional support provided at the University School of Automation and Robotics, GGSIPU East Delhi Campus related to research in artificial intelligence and cybersecurity.

8. REFERENCES

[1] M Ahmed, A Mahmood, and J Hu. Network traffic anomaly detection using machine learning. *Journal of Network and Computer Applications*, 215:103678, 2023.

[2] Francesco Blefari, Cristian Cosentino, Francesco Aurelio Pironti, Angelo Furfaro, and Fabrizio Marozzo. Cyberrag: An agentic rag cyber attack classification and reporting tool. *Future Generation Computer Systems*, page 108186, 2025.

[3] A Buczak and E Guven. A survey of data mining and machine learning methods for cybersecurity intrusion detection. *IEEE Communications Surveys & Tutorials*, 18(2):1153–1176, 2016.

[4] D Denning. An intrusion detection model. *IEEE Transactions on Software Engineering*, SE-13(2):222–232, 1987.

[5] P Garcia-Teodoro, J Diaz-Verdejo, G Macia-Fernandez, and E Vazquez. Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers & Security*, 28(1-2):18–28, 2009.

[6] Firuz Kamalov, Sherif Moussa, Rita Zgheib, and Omar Mashaal. Feature selection for intrusion detection systems. In *2020 13th International Symposium on Computational Intelligence and Design (ISCID)*, pages 265–269, 2020.

[7] S Kasongo and Y Sun. Performance analysis of intrusion detection systems using feature selection on the unsw-nb15 dataset. *Journal of Big Data*, 7(1):1–20, 2020.

[8] G Kim, S Lee, and S Kim. A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. *Expert Systems with Applications*, 41(4):1690–1700, 2014.

[9] P Lewis, E Perez, A Piktus, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474, 2020.

[10] Hongjuan Li, Hui Kang, Jiahui Li, Geng Sun, Ruichen Zhang, Jiacheng Wang, Dusit Niyato, Wei Ni, and Abbas Jamalipour. Multi-agent collaborative intrusion detection for low-altitude economy iot: An llm-enhanced agentic ai framework. *arXiv preprint arXiv:2601.17817*, 2026.

[11] Zong-Xun Li, Yu-Jun Li, Yi-Wei Liu, Cheng Liu, and Nan-Xin Zhou. K-ctiaa: Automatic analysis of cyber threat intelligence based on a knowledge graph. *Symmetry*, 15(2):337, 2023.

[12] R Lippmann, J Haines, D Fried, J Korba, and K Das. The 1999 darpa off-line intrusion detection evaluation. *Computer Networks*, 34(4):579–595, 2000.

[13] Xiang Luo, Chang Liu, Gang Xiong, Chen Yang, Gaopeng Gou, Yaochen Ren, and Zhen Li. Malrag: A retrieval-augmented llm framework for open-set malicious traffic identification. *arXiv preprint arXiv:2511.14129*, 2025.

[14] N Moustafa and Jill Slay. The unsw-nb15 dataset for network intrusion detection systems. In *Military Communications and Information Systems Conference*, pages 1–6, 2015.

[15] S Russell and P Norvig. *Artificial Intelligence: A Modern Approach*. Pearson, 4th edition, 2021.

[16] I Sharafaldin, A Lashkari, and A Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *International Conference on Information Systems Security and Privacy*, pages 108–116, 2018.

[17] N Shone, T Ngoc, V Phai, and Q Shi. Deep learning approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(3):234–245, 2022.

[18] Marco Simoni, Andrea Saracino, Vinod P, and Mauro Conti. Morse: Bridging the gap in cybersecurity expertise with retrieval augmented generation. In *Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing*, pages 1213–1222, 2025.

[19] R Sommer and V Paxson. Outside the closed world: On using machine learning for network intrusion detection. In *IEEE Symposium on Security and Privacy*, pages 305–316, 2010.

[20] M Tavallaee, E Bagheri, W Lu, and A Ghorbani. A detailed analysis of the kdd cup 99 data set. In *IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pages 1–6, 2009.

[21] W Wang, Y Sheng, J Wang, et al. Hast-ids: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection. *IEEE Transactions on Information Forensics and Security*, 17:1234–1247, 2022.

[22] C Yin, Y Zhu, J Fei, and X He. A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access*, 5:21954–21961, 2017.