

RAG-NIDS+: Temporally Aware Retrieval-Augmented Network Intrusion Detection and Response

Anh Tran Hai¹, Abdelhamid Mellouk^{2,*}

¹School of Information and Communication Technology, Hanoi University of Science and Technology, Ha Noi, Vietnam

²TincNET, LISSI, University of Paris-Est Creteil, Vitry-sur-Seine, F-94400, France

*Corresponding author email: mellouk@u-pec.fr

Abstract

Network intrusion detection systems should not only detect anomalous traffic but also provide relevant evidence and actionable response guidance. This paper presents RAG-NIDS+, an end-to-end framework integrating enhanced anomaly detection, temporally aware threat-intelligence retrieval, and evidence-grounded recommendation generation. A temporally attentive Long Short-Term Memory (LSTM) autoencoder combines attention-weighted sequence representation, dynamics-aware reconstruction, and hybrid anomaly scoring to capture both sustained and short-lived anomalies. Hybrid lexical-semantic retrieval with time-decay weighting prioritizes relevant and recent threat intelligence, which is then combined with anomaly metadata and operational context to ground LLaMA-3-8B-Instruct. Retrieval parameters are selected on validation data and fixed before testing. Four experts assess recommendations across 120 scenarios for correctness, relevance, clarity, and actionability, with independent hallucination judgments followed by consensus adjudication. Experiments on UNSW-NB15, CICIDS2017, and TON-IoT achieved Macro-F1 scores of 0.894, 0.962, and 0.893, respectively. Temporal weighting improved Precision@10 from 0.848 to 0.910 and nDCG@10 from 0.862 to 0.920. Compared with the ungrounded language model, the full framework reduced hallucination from 28.4% to 3.2% and achieved an overall human score of 4.58 out of 5. On the evaluated A100-based server, RAG-NIDS+ achieved a P95 end-to-end latency of 425 ms, supporting analyst-facing intrusion analysis and response generation.

Keywords: Evidence-grounded recommendation generation, network intrusion detection system (NIDS), retrieval-augmented generation (RAG), threat intelligence.

1. Introduction

Network intrusion detection is only the first step in incident handling. After an alert is raised, analysts must determine its significance, identify supporting threat intelligence, and select actions appropriate to the affected assets. Signature-based systems recognize known patterns, while anomaly-based detectors identify deviations from normal traffic; however, an anomaly score alone provides limited evidence for these operational decisions [1]. This motivates connecting detection with evidence retrieval and contextual response support.

Deep learning and attention-based models improve traffic modeling [2, 3], while retrieval-augmented generation (RAG) grounds language-model outputs in external knowledge [4, 5]. Recent frameworks, including RLFE-IDS and CyberRAG, demonstrate the relevance of RAG and large language models to intrusion detection and attack analysis [6, 7]. Nevertheless, retrieving topically relevant information does not necessarily provide temporally appropriate evidence. An older advisory may describe the same attack family but contain superseded indicators or mitigation guidance. Conversely, prioritizing recency

too strongly can discard useful historical knowledge. Response generation therefore requires a balance between relevance and recency, together with the detected event and operational context.

We propose **RAG-NIDS+** to investigate this detection-to-response problem. A temporally attentive LSTM autoencoder (TA-LSTM-AE) combines attention-weighted representations, dynamics-aware reconstruction, and hybrid anomaly scoring. Detected events trigger hybrid lexical-semantic cyber threat intelligence (CTI) retrieval, with timestamp filtering and time-decay weighting. Retrieved evidence, anomaly metadata, and asset-specific constraints then ground LLaMA-3-8B-Instruct for analyst-facing recommendations. The methodological contribution lies in the detector's combined representation, training objective, and anomaly score, together with explicit timestamp-constrained, recency-aware evidence ranking for downstream response support. Attention, BM25/dense retrieval, exponential decay, and RAG are established building blocks; we do not claim their invention or a new language-model architecture.

The main contributions are as follows:

p-ISSN 3093-3331

e-ISSN 3093-3323

<https://doi.org/10.51316/jst.xxx.dsai.xxxx.xx.x.x>

Received: Aug 25, 2026; Revised: Oct 7, 2026;

Accepted: Oct 10, 2026; Online: Oct 21, 2026

- A **normal-only temporal anomaly detector** combining attention-weighted encoder states, a temporal-dynamics reconstruction objective, and mean-quantile anomaly scoring, with component-wise ablation.
- An **event-conditioned CTI ranking scheme** combining lexical-semantic relevance with timestamp eligibility and an additive recency preference, evaluated against retrieval without temporal decay.
- An **analyst-facing detection-to-response framework** that conditions a pretrained Large Language Model (LLM) on anomaly metadata, retrieved evidence, and operational constraints, with separate evaluations of detection, retrieval, recommendation quality, and runtime.

The remainder of this paper reviews related work in Section 2, presents the methodology in Section 3, reports the evaluation in Section 4, and concludes in Section 5.

2. Related Work

We review detection models, CTI integration, and recent RAG-based security frameworks to position the proposed detection-to-response pipeline.

2.1. Recent Deep Learning Approaches in NIDS

Deep models learn traffic representations beyond manually specified signatures. Recent studies investigate industrial-IoT detection [2], chaotic optimization for deep NIDS [8], and Transformer-based cloud intrusion detection [9]. In 2025, Umer *et al.* [3] combine wrapper-based feature selection with a multi-head attention Transformer for intrusion classification on UNSW-NB15. These studies advance traffic classification, but detection performance alone does not establish the quality of retrieved evidence or subsequent response guidance. RAG-NIDS+ therefore evaluates detection, CTI retrieval, and recommendation quality separately.

2.2. Threat Intelligence and RAG-Based Security Frameworks

Sarhan *et al.* [10] investigate federated CTI sharing, while ORISHA [11] couples collaborative intelligence with intrusion detection. Recent RAG-based frameworks pursue different tasks. RLFE-IDS [6] combines RAG with the FE-Net embedding model for intrusion detection. TechniqueRAG [12] focuses on adversarial-technique annotation in CTI text, using instruction-tuned generation and LLM reranking. CyberRAG [7] uses an LLM agent to orchestrate specialized classifiers and iterative retrieval and reasoning, including refinement of threat labels and explanations. In RAG-NIDS+, the normal-only detector makes the anomaly decision before CTI retrieval, and generation does not revise that decision. Its

specific focus is the combined detector formulation and timestamp-constrained, recency-aware evidence ranking for operational recommendations. These distinctions concern task and method design, not demonstrated performance superiority over the cited frameworks.

2.3. Identified Challenges and Motivation

RAG-NIDS+ addresses two method-design questions: how a normal-only reconstruction detector can combine sequence representation, temporal-dynamics preservation, and localized-error scoring; and how event-relevant CTI can be ranked using both topical relevance and evidence age. The downstream generator applies retrieved evidence to operational response support rather than determining the anomaly label. We assess the detector extensions through sequential ablation and temporal ranking through comparison with retrieval without decay. This positions the contribution as a specific extension and evaluation of established techniques, rather than the generic combination of an IDS and a language model. The framework does not introduce joint end-to-end optimization or a new LLM architecture.

3. Methodology

3.1. Problem Definition

Let $\mathbf{X} = \{x_1, x_2, \dots, x_T\}$ denote a sequence of network-traffic observations, where each x_t is a d -dimensional feature vector. The objective is to detect anomalous behavior and transform each detected event into an evidence-grounded response. The anomaly detector f_θ maps $\mathbf{X}$ to an anomaly representation $\mathbf{A} = (y, s, \mathbf{m})$, where y is the predicted label, s is the anomaly score, and $\mathbf{m}$ contains the associated event metadata. Given $\mathbf{A}$, a threat-intelligence knowledge base $\mathcal{K}$, and query time t_q , the retrieval function g_ϕ returns the top- k documents TI_k , ranked according to their contextual and temporal relevance. Finally, the response generator h_ψ combines $\mathbf{A}$, TI_k , and the operational context c to produce a set of actionable recommendations $\mathcal{R}$. Thus, the overall processing pipeline is expressed as $\mathbf{X} \rightarrow \mathbf{A} \rightarrow \text{TI}_k \rightarrow \mathcal{R}$.

3.2. Proposed Framework: RAG-NIDS+

The proposed framework RAG-NIDS+ operates as a unified pipeline, transforming raw network traffic into actionable insights through a sequence of well-defined modules.

As illustrated in Fig. 1, the proposed framework begins with the **Data Preprocessing** module, which transforms raw network traffic into structured datasets suitable for further analysis. These structured datasets are then analyzed by the **Anomaly Detection** module, which leverages an TA-LSTM-AE detector to identify sequential patterns in the network traffic. An anomaly score is computed to effectively detect deviations from normal behavior. Once anomalies are detected, they are

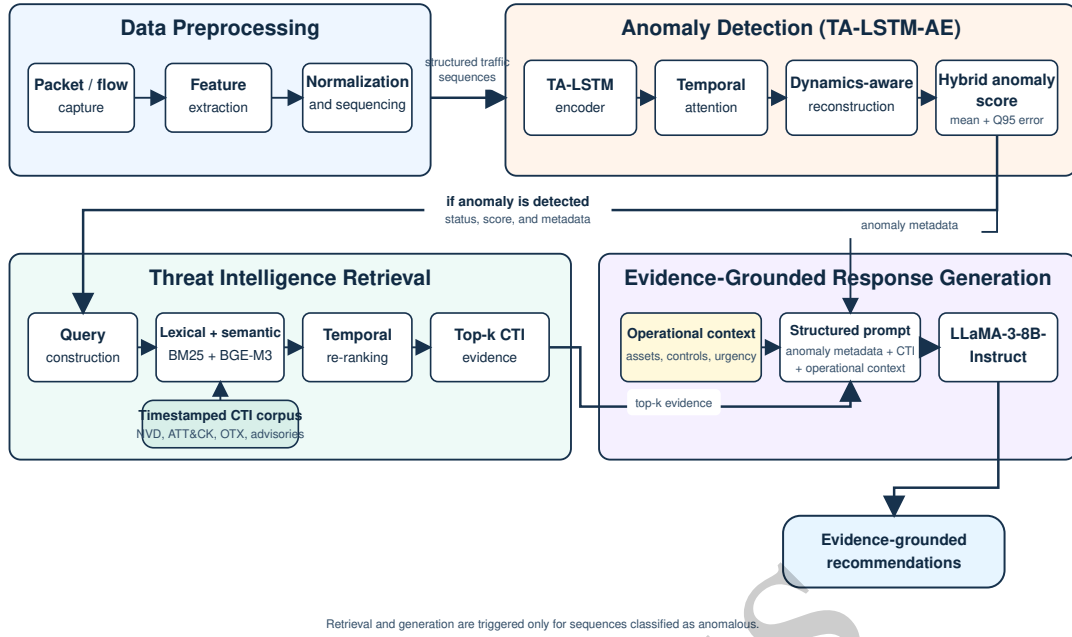


Fig. 1. RAG-NIDS+ framework

forwarded to the **Retrieval Module**, which retrieves relevant and temporally-aware threat intelligence from both internal and external knowledge sources. Finally, the **Response Generation** module generates recommendations by incorporating the severity of the detected anomaly, system-specific configurations, and real-time operational constraints. The roles and mechanisms of these four modules are detailed in the subsequent subsections.

3.3. Data Preprocessing Module

The Data Preprocessing Module converts raw network traffic into temporal sequences suitable for anomaly detection. Network flows are first reconstructed, after which statistical and protocol-related features, including packet and byte counts, port numbers, protocol types, flow durations, and timing information, are extracted. Categorical attributes are transformed using one-hot encoding, while continuous attributes are standardized using the mean and standard deviation computed exclusively from the training partition to prevent data leakage. Finally, the resulting feature vectors are ordered chronologically and grouped into fixed-length sequences, which are subsequently provided to the TA-LSTM-AE detector.

3.4. Anomaly Detection Module

TA-LSTM-AE adapts an LSTM autoencoder with additive temporal attention, temporal-dynamics reconstruction, and mean-quantile scoring; these components are not individually claimed as novel. Training uses normal traffic; Table 7 reports component-wise evaluation.

Encoder and temporal attention. For $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_T)$, $T > 1$, and $\mathbf{x}_t \in \mathbb{R}^d$, the encoder produces $\mathbf{h}_t \in \mathbb{R}^H$. Attention computes

$$u_t = \mathbf{v}^\top \tanh(\mathbf{W}_h \mathbf{h}_t + \mathbf{b}_h), \quad a_t = \frac{\exp(u_t)}{\sum_{j=1}^T \exp(u_j)}$$

where $\mathbf{W}_h \in \mathbb{R}^{d_a \times H}$ and $\mathbf{b}_h, \mathbf{v} \in \mathbb{R}^{d_a}$ are jointly learned parameters. The shared vector $\mathbf{v}$ produces scalar scores rather than acting as an external query. With $a_t \geq 0$ and $\sum_t a_t = 1$, the latent representation is

$$\mathbf{z} = \sum_{t=1}^T a_t \mathbf{h}_t \in \mathbb{R}^H$$

It aggregates all states rather than only the final state; input-dependent attention weights are neither attack labels nor guaranteed anomaly explanations.

Decoder and training objective. The decoder receives $\mathbf{z}$ repeated over T timesteps, with zero initial hidden/cell states in all layers. A shared linear projection maps outputs to $\hat{\mathbf{x}}_t \in \mathbb{R}^d$, forming $\hat{\mathbf{X}}$. Training minimizes

$$\mathcal{L}_{\text{rec}} = \frac{1}{T} \sum_{t=1}^T \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2^2$$

$$\mathcal{L}_{\text{dyn}} = \frac{1}{T-1} \sum_{t=2}^T \|(\mathbf{x}_t - \mathbf{x}_{t-1}) - (\hat{\mathbf{x}}_t - \hat{\mathbf{x}}_{t-1})\|_2^2$$

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \eta \mathcal{L}_{\text{dyn}}$$

where $\eta \geq 0$ balances feature reconstruction and preservation of first-order temporal differences. All parameters are optimized jointly on normal sequences.

Inference and anomaly scoring. For $e_t = \|x_t - \hat{x}_t\|_2^2$, the sequence score is

$$S(\mathbf{X}) = \rho \left(\frac{1}{T} \sum_{t=1}^T e_t \right) + (1 - \rho) Q_q(e_1, \dots, e_T)$$

where $\rho \in [0, 1]$ balances sustained deviations and elevated errors captured by the empirical quantile Q_q , with $q = 0.95$. Loss/scoring settings use validation data; threshold τ uses normal validation sequences. Settings and model parameters are fixed before testing. When $S(\mathbf{X}) > \tau$, the score and event metadata trigger CTI retrieval.

3.5. Threat Intelligence Retrieval Module

Queries use anomaly scores and event metadata for lexical-semantic CTI retrieval. Only documents published by query time t_q are eligible. For publication time t_i , temporal relevance is

$$T(D_i) = e^{-\lambda(t_q - t_i)}$$

Age is measured in days; $\lambda \geq 0$ has units day^{-1} . Zero removes age preference; larger values favor recency. Timestamp filtering independently excludes future documents. Candidates receive

$$R(D_i) = \alpha R_{\text{lex}}(D_i) + \beta R_{\text{sem}}(D_i) + \gamma T(D_i)$$

where R_{lex} is normalized lexical relevance, R_{sem} is embedding similarity, and nonnegative weights sum to one. The top- k ranked documents, TI_k , support generation.

Writing $R_{\text{rel}} = \alpha R_{\text{lex}} + \beta R_{\text{sem}}$, two eligible documents satisfy

$$R(D_i) - R(D_j) = R_{\text{rel}}(D_i) - R_{\text{rel}}(D_j) + \gamma[T(D_i) - T(D_j)]$$

Recency can outweigh a relevance disadvantage while preserving competitive historical evidence. At $\lambda = 0.5 \text{ day}^{-1}$, the temporal-score half-life is approximately 1.39 days.

3.6. Response Generation Module

The Response Generation Module converts each detected anomaly and its retrieved threat intelligence into an actionable response using LLaMA-3-8B-Instruct. Its inputs comprise the anomaly representation $\mathbf{A}$, the top- k retrieved documents TI_k , and operational context $\mathbf{c}$, such as asset configuration, available security controls, and response urgency. These inputs are organized into a structured prompt containing an anomaly summary, supporting evidence with source and timestamp information, and the required response format. The model is instructed to assess severity, explain the threat using the supplied evidence, recommend prioritized mitigation actions, and avoid unsupported claims. The generation process is expressed as

$$\mathcal{R} = \text{LLaMA-3-8B-Instruct}(\text{Prompt}(\mathbf{A}, \text{TI}_k, \mathbf{c}))$$

where $\mathcal{R}$ denotes the generated recommendation. Because the prompt is conditioned on event-specific metadata, recent threat intelligence, and system constraints, the resulting response adapts to the detected incident while remaining grounded in retrieved evidence. This module supports analysis and response generation but does not modify the anomaly detector's decision.

3.7. Computational Complexity Analysis

We separate online inference from offline CTI indexing. Let T, d_x, H, d_a, ℓ_a denote sequence length, feature dimension, LSTM hidden size, attention dimension, and recurrent-layer count; N, d_e, C, k denote corpus size, embedding dimension, candidate count, and retrieval depth.

Anomaly detection: LSTM encoding/decoding and attention require $\mathcal{O}(T(Hd_x + \ell_a H^2))$ and $\mathcal{O}(THd_a)$, respectively. Reconstruction costs $\mathcal{O}(Td_x)$; quantile selection costs $\mathcal{C}_q(T) = \mathcal{O}(T)$ in expectation or $\mathcal{O}(T \log T)$ with sorting. Thus,

$$\mathcal{C}_{\text{det}} = \mathcal{O}(T(Hd_x + \ell_a H^2 + Hd_a) + \mathcal{C}_q(T))$$

For $d_a = \mathcal{O}(H)$, attention preserves the dominant LSTM complexity. The temporal-dynamics loss adds $\mathcal{O}(Td_x)$ only during training. Activation memory is $\mathcal{O}(TH)$ for inference and $\mathcal{O}(\ell_a TH)$ for training, excluding input/output storage.

Threat-intelligence retrieval: With offline corpus embedding and indexing, query encoding, BM25, ANN search, and candidate reranking yield

$$\begin{aligned} \mathcal{C}_{\text{ret}} = & \mathcal{C}_{\text{enc}}(L_q) + \mathcal{O}(P_q) \\ & + \mathcal{C}_{\text{ANN}}(N, d_e) + \mathcal{O}(Cd_e + C \log k) \end{aligned}$$

where L_q is query length and P_q counts visited postings. ANN cost depends on the index and search parameters; exhaustive search costs $\mathcal{O}(Nd_e)$. Temporal scoring adds only $\mathcal{O}(C)$.

Response generation: For P prompt tokens, G generated tokens, hidden dimension D , and ℓ_g transformer layers, prefill and decoding with key-value caching cost

$$\mathcal{C}_{\text{prefill}} = \mathcal{O}(\ell_g(PD^2 + P^2D))$$

$$\mathcal{C}_{\text{decode}} = \mathcal{O}(\ell_g(GD^2 + (PG + G^2)D))$$

Hence, $\mathcal{C}_{\text{gen}} = \mathcal{C}_{\text{prefill}} + \mathcal{C}_{\text{decode}}$. Cache memory is $\mathcal{O}(\ell_g(P + G)D)$, excluding model parameters. Bounding retrieval depth, chunk length, and output length controls generation cost.

End-to-end complexity: Retrieval and generation run only when the anomaly score exceeds threshold τ :

$$\mathcal{C}_{\text{total}} = \mathcal{C}_{\text{det}} + \mathbf{1}[S(\mathbf{X}) > \tau](\mathcal{C}_{\text{ret}} + \mathcal{C}_{\text{gen}})$$

Table 1. Datasets and threat-intelligence corpus used in the evaluation

Source	Features	Train	Val.	Test
<i>Intrusion Detection Traffic (Records)</i>				
UNSW-NB15	49	131,506	43,835	82,332
CICIDS2017	80	1,540,210	385,052	641,755
TON-IoT	44	1,200,500	300,125	500,210
<i>Threat-Intelligence Corpus (CTI)</i>				
Total Documents			142,580	
Snapshot Date			Jan 15, 2026	

For anomalous-sequence fraction p_a , the expected cost is

$$\mathbb{E}[\mathcal{C}_{\text{total}}] = \mathcal{C}_{\text{det}} + p_a(\mathcal{C}_{\text{ret}} + \mathcal{C}_{\text{gen}})$$

In practice, LLM generation dominates online cost, while detection remains lightweight and retrieval uses precomputed indexes.

4. Experiments and Evaluation

4.1. Experimental Protocol

4.1.1. Datasets and threat-intelligence corpus

Table 1 summarizes the datasets. Preprocessing removed duplicate/invalid records, one-hot encoded categories, and standardized features using training statistics. Shared flows, sessions, or capture periods remained within one partition.

CTI from OpenCTI/OTX included NVD, MITRE ATT&CK, and vendor advisories. BGE-m3/Milvus indexed 512-token chunks with 64-token overlap, retaining parent IDs, sources, and timestamps.

The 300 queries (100/dataset) represented anomalous 32-observation sequences, using protocol, service/port, behavior, and score; labels served only stratification/analysis. Relevance reflected behavioral/technical correspondence, not assumed vulnerability/campaign identities.

Three experts independently graded pooled, deduplicated top-20 documents from BM25, dense, hybrid, and temporal retrieval: 0 irrelevant, 1 partial, 2 directly supportive; disagreements reached consensus. Dataset/attack-family-stratified validation/test splits contained 60/240 queries (20/80 per dataset), keeping overlapping events together; test judgments never informed tuning.

Queries used January 15, 2026 for retrospective analysis, not historical replay. Later CTI was excluded; ages used days. Each parent retained its highest-scoring chunk. Test-query macro-averages used grades > 0 for P@10/MRR, gains $2^s - 1$ for nDCG@10, and the top-10 proportion published within 30 days for freshness.

4.1.2. Implementation and baselines

The detector uses $T = 32$, two-layer LSTM encoder/decoder networks with $H = H_{\text{dec}} = 64$, and $d_a = 32$. Attention aggregates encoder states; training combines reconstruction/dynamics losses, and scoring combines mean/95th-percentile errors. Loss weights and thresholds were validation-selected and fixed for testing.

Retrieval combines BM25, BGE-m3 cosine similarity, and temporal reranking with validation-selected normalized weights. Sensitivity sweeps varied λ at $k = 10$ and k at $\lambda = 0.5$, otherwise holding settings fixed (Fig. 2). Validation nDCG@10, supplemented by P@10, MRR, and freshness, selected $\lambda = 0.5$, $k = 10$ for testing. LLaMA-3-8B-Instruct used temperature 0.2, sampling top- $k = 50$, and a 512-token output limit. Serving used vLLM/BF16 on an NVIDIA A100.

We implemented RF [13], AE [14], LSTM-AE [15], and 1D-CNN [16] with common partitions, preprocessing, and validation tuning, retaining their learning regimes. LSTM-AE matched TA-LSTM-AE’s dimensions, optimizer, and threshold selection, but used final-state encoding, reconstruction MSE, and mean-error scoring. Retrieval baselines comprised BM25, dense, and non-temporal hybrid retrieval; recommendation baselines comprised templates, LLM-only, and conventional RAG.

The wrapper-based feature-selection/multi-head attention Transformer of Umer *et al.* [3] was implemented on UNSW-NB15 under the same protocol/metrics. Feature selection used training data; tuning, checkpoints, and thresholds used validation data. It used labeled training versus TA-LSTM-AE’s normal-only training. Detection reports mean $\pm$ standard deviation over five seeds.

Experiments used Intel Xeon, 256 GB RAM, and NVIDIA A100 hardware. Metrics covered detection (Macro-F1, PR-AUC, FPR), retrieval (P@10, MRR, nDCG@10, freshness), and recommendations (BERTScore, human ratings, hallucination). Human evaluation was randomized/blinded; agreement used Krippendorff’s α .

4.2. Results and Discussion

4.2.1. Intrusion-detection performance

Table 2 compares our implementations using common partitions and preprocessing within each dataset. Table 3 adds the adapted UNSW-NB15 detector of Umer *et al.* [3]; results are not imported from the literature.

TA-LSTM-AE improves Macro-F1 over the strongest baseline by 3.6, 2.7, and 5.2 percentage points on UNSW-NB15, CICIDS2017, and TON-IoT, respectively, with relative FPR reductions of 34.6%, 42.9%, and 37.7%. Fewer false alerts can reduce

Table 2. Binary intrusion-detection performance obtained from our implementations under common dataset partitions and preprocessing (mean $\pm$ standard deviation over five runs)

Method	UNSW-NB15			CICIDS2017			TON-IoT		
	Macro-F1	PR-AUC	FPR	Macro-F1	PR-AUC	FPR	Macro-F1	PR-AUC	FPR
RF	0.842 $\pm$ 0.005	0.865 $\pm$ 0.004	0.068 $\pm$ 0.003	0.921 $\pm$ 0.003	0.945 $\pm$ 0.002	0.028 $\pm$ 0.002	0.825 $\pm$ 0.006	0.848 $\pm$ 0.005	0.075 $\pm$ 0.004
AE	0.815 $\pm$ 0.006	0.832 $\pm$ 0.005	0.084 $\pm$ 0.004	0.884 $\pm$ 0.005	0.902 $\pm$ 0.004	0.045 $\pm$ 0.003	0.798 $\pm$ 0.007	0.815 $\pm$ 0.006	0.092 $\pm$ 0.005
Conventional LSTM-AE	0.858 $\pm$ 0.004	0.881 $\pm$ 0.003	0.052 $\pm$ 0.002	0.935 $\pm$ 0.003	0.958 $\pm$ 0.002	0.021 $\pm$ 0.001	0.841 $\pm$ 0.005	0.867 $\pm$ 0.004	0.061 $\pm$ 0.003
ID-CNN	0.851 $\pm$ 0.005	0.874 $\pm$ 0.004	0.058 $\pm$ 0.003	0.928 $\pm$ 0.004	0.951 $\pm$ 0.003	0.025 $\pm$ 0.002	0.836 $\pm$ 0.005	0.859 $\pm$ 0.005	0.066 $\pm$ 0.004
Proposed TA-LSTM-AE	0.894 $\pm$ 0.003	0.918 $\pm$ 0.002	0.034 $\pm$ 0.002	0.962 $\pm$ 0.002	0.981 $\pm$ 0.001	0.012 $\pm$ 0.001	0.893 $\pm$ 0.003	0.922 $\pm$ 0.003	0.038 $\pm$ 0.002

Table 3. Additional UNSW-NB15 comparison (mean $\pm$ standard deviation over five runs). The adapted Umer *et al.* baseline uses supervised training; TA-LSTM-AE uses normal-only training.

Method	Macro-F1	PR-AUC	FPR
Umer <i>et al.</i> (2025), adapted	0.865 $\pm$ 0.003	0.882 $\pm$ 0.003	0.051 $\pm$ 0.002
Proposed TA-LSTM-AE	0.894 $\pm$ 0.003	0.918 $\pm$ 0.002	0.034 $\pm$ 0.002

Table 4. CTI retrieval performance on held-out event-derived queries, evaluated at the document level using expert relevance judgments.

Method	P@10	MRR	nDCG@10	Fresh.
BM25	0.742	0.785	0.761	0.618
Dense retrieval	0.814	0.852	0.828	0.654
Hybrid, no temporal decay	0.848	0.881	0.862	0.683
RAG-NIDS+ temporal retrieval	0.910	0.940	0.920	0.853

downstream processing. These gains agree with the ablation of attention, temporal-dynamics loss, and mean-quantile scoring. The larger TON-IoT gain remains dataset-specific; retrieval and generation occur after detection and do not explain these improvements.

Against adapted Umer *et al.* [3], TA-LSTM-AE raises Macro-F1 from 0.865 to 0.894 and PR-AUC from 0.882 to 0.918, while lowering FPR from 0.051 to 0.034: a 2.9-percentage-point Macro-F1 gain and 33.3% relative FPR reduction. This may reduce downstream workload, but does not isolate architectural effects. The comparison covers only UNSW-NB15 and contrasts supervised Transformer training with normal-only TA-LSTM-AE training.

4.2.2. Temporally aware threat-intelligence retrieval

Table 4 compares lexical, semantic, hybrid, and temporal retrieval on the same corpus and queries, isolating temporal weighting against the hybrid baseline.

Temporal weighting promotes updated CTI over similarly relevant older evidence, potentially limiting reliance on superseded information. Rapid daily decay mainly affects recent documents; historical evidence remains retrievable through lexical and semantic relevance. Recency, however, does not guarantee correctness.

Compared with hybrid retrieval, P@10 rises from 0.848 to 0.910, nDCG@10 from 0.862 to 0.920, and MRR from 0.881 to 0.940, alongside higher freshness. Hallucination decreases from 9.6% for conventional RAG to 3.2%, consistent with improved evidence

selection. The decrease from 28.4% for LLM-only generation also reflects retrieval grounding. Aggregate results neither attribute individual errors to stale CTI nor establish recency as factual accuracy.

In Fig. 2, $\lambda = 0.5$ maximizes plotted validation relevance metrics at $k = 10$; weaker decay retains older evidence, while stronger decay risks suppressing relevant history. At $\lambda = 0.5$, $k = 10$ maximizes nDCG@10: smaller depths limit evidence, whereas larger depths reduce relevance/freshness and enlarge context. These validation-selected values balance relevance, recency, and context size without implying universal optimality.

4.2.3. Recommendation quality

Four experts rated 120 scenarios with identical metadata/context; outputs were anonymized/randomized. LLM-only used no evidence, conventional RAG used hybrid retrieval, and RAG-NIDS+ used temporal retrieval.

Correctness, relevance, clarity, and actionability measured factual consistency, contextual fit, comprehensibility, and feasible prioritized actions (1–5: deficient to satisfactory). Scores averaged experts/scenarios; overall scores equally weighted unrounded criterion means.

Hallucination meant unsupported/contradicted factual assertions, excluding conditional hypotheses, precautions, or low actionability alone. Experts judged independently, resolving disagreements by consensus; flagged outputs counted once.

RAG-NIDS+ achieved 4.58 overall, 3.2% hallucination versus LLM-only’s 28.4%, and +0.66 actionability over conventional RAG. Pre-adjudication criterion-specific ordinal Krippendorff’s α averaged 0.81 across method-scenario pairs.

Table 6 presents constructed cases with hypothetical CTI.

At $\lambda = 0.5 \text{ day}^{-1}$, Case A’s temporal scores (0.607, 1.69×10^{-10}) favor newer guidance under similar relevance; Case B shows unsupported inference despite recent evidence.

LLM-only evaluates pipeline gains; conventional RAG better isolates temporal weighting. CTI gaps and unsupported generation persist; constructed cases cannot causally explain aggregate gains.

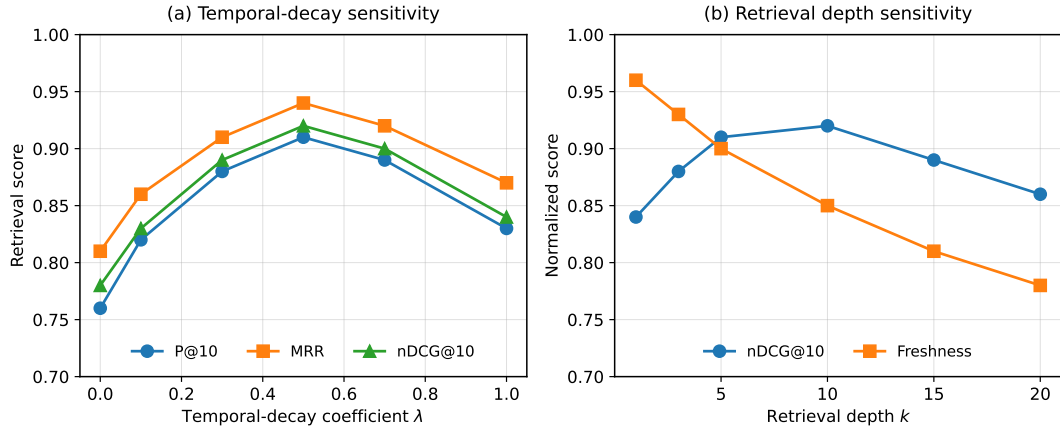


Fig. 2. Validation sensitivity to (a) the temporal-decay coefficient λ at $k = 10$, and (b) retrieval depth k at $\lambda = 0.5$. Other retrieval settings are held fixed

Table 5. Recommendation quality. Human scores use a 1–5 scale; lower hallucination is better

Method	BERTScore	Correctness	Actionability	Hallucination (%)	Overall human score
Static templates	0.712	2.85	3.10	0.0	2.70
LLM only	0.805	3.20	3.15	28.4	3.08
Conventional RAG	0.864	4.12	3.90	9.6	4.02
Full RAG-NIDS+	0.898	4.65	4.56	3.2	4.58

Table 6. Constructed qualitative cases with shortened response excerpts. Both use January 15, 2026 as the query date

Item	Evidence and response
Case A: potential temporal-retrieval benefit	
Anomaly	One source scans multiple server ports; legitimate access must remain available.
CTI evidence	A1 (Dec 1, 2025): investigate reconnaissance; block unauthorized sources. A2 (Jan 14, 2026): verify authorization; use scoped controls. Illustrative A1/A2 ranks: conventional 1/3; temporal 3/1.
LLM-only	“The server is compromised; isolate it and block the source.”
Conventional RAG	“Likely port scanning; block the source and disable unnecessary services.”
RAG-NIDS+	“Possible reconnaissance, not confirmed compromise. Verify authorization, inspect logs, and consider rate-limiting untrusted traffic while preserving approved access.”
Interpretation	Updated guidance supports conditional, context-aware actions.
Case B: failure despite recent evidence	
Anomaly	Repeated SSH login failures; successful-login and endpoint evidence unavailable.
CTI evidence	B1 (Jan 13, 2026): investigate brute force, review logs, and apply warranted controls; compromise is not established.
RAG-NIDS+	“Brute force compromised the server; isolate it immediately.”
Failure analysis	Login failures do not prove compromise; isolation lacks supporting context.

4.2.4. Ablation and runtime performance

Table 7 incrementally adds detector enhancements, then retrieval and generation. P50/P95 latency uses 1,000 requests after 100 warm-ups. Anomalous requests were processed sequentially, timed from preprocessed-sequence submission through completed recommendation, including detection, retrieval, prompting, and generation but excluding collection, preprocessing, and model/index loading.

Attention, dynamics-aware loss, and hybrid scoring increase average Macro-F1 by 1.6, 1.3, and 0.9 percentage points in the tested order; these are not independent effects. Downstream modules leave detection unchanged. Temporal retrieval adds 0.058 nDCG@10, raising P95 from 68.0 to 72.5 ms. Temporal RAG raises human scores from 4.02 to 4.58 and lowers hallucination from 9.6% to 3.2%, while P95 rises from 410 to 425 ms. Deployment requirements and latency limitations appear in Section 4.2.5.

4.2.5. Practical deployment and limitations

Latency was measured on an Intel Xeon server (256 GB RAM, NVIDIA A100), excluding setup overhead, and does not characterize edge deployment. P95 latencies reached 18.5 ms for detection, 72.5 ms with temporal retrieval, and 425 ms end-to-end. Operational deployment requires dedicated memory, maintained CTI indexes, and concurrency limits; asynchronous execution remains unassessed.

Real-time guarantees and performance under sustained or burst loads remain unverified. Key limitations include evaluation on offline traces, restriction to the UNSW-NB15 baseline, CTI freshness dependence, and a 120-scenario human study. Future work will address sustained loads, broader baselines, cross-domain transfer, and noisy CTI.

Finally, residual hallucinations mandate analyst verification before executing disruptive actions, particularly when host or authentication telemetry is missing.

Table 7. Component ablation and runtime under matched settings. Avg. Macro-F1 is the unweighted mean across UNSW-NB15, CICIDS2017, and TON-IoT.

Configuration	Avg. Macro-F1	nDCG@10	Human score	Halluc. (%)	P50 (ms)	P95 (ms)
Conventional LSTM-AE	0.878	–	–	–	–	–
+ Temporal attention	0.894	–	–	–	–	–
+ Dynamics-aware loss	0.907	–	–	–	–	–
+ Hybrid anomaly score (proposed detector)	0.916	–	–	–	12.4	18.5
Proposed detector + hybrid retrieval	0.916	0.862	–	–	45.2	68.0
Proposed detector + temporal retrieval	0.916	0.920	–	–	48.6	72.5
Proposed detector + conventional RAG	0.916	0.862	4.02	9.6	285.0	410.0
Full RAG-NIDS+	0.916	0.920	4.58	3.2	292.0	425.0

5. Conclusion

RAG-NIDS+ integrates intrusion detection, temporal CTI retrieval, and grounded recommendations. Detector Macro-F1 reached 0.894/0.962/0.893 on UNSW-NB15/CICIDS2017/TON-IoT, outperforming evaluated baselines, including adapted Umer *et al.* [3] on UNSW-NB15. Temporal retrieval raised nDCG@10 from 0.862 to 0.920; grounded generation achieved a human score of 4.58 and reduced hallucination from 28.4% to 3.2%. A100-server P95 latency was 425 ms; hard real-time and sustained-load performance remain unverified. Future work targets cross-domain generalization, noisy/poisoned CTI, and sustained-traffic deployment.

References

- [1] O. H. Abdulganiyu, T. A. Tchakoucht, and Y. K. Saheed, Towards an efficient model for network intrusion detection system (IDS): systematic literature review, *Wireless Networks*, vol. 30, iss. 1, pp. 453–482, Sep. 2023. <https://doi.org/10.1007/s11276-023-03495-2>
- [2] H. Nandanwar and R. Katarya, Deep learning enabled intrusion detection system for industrial IoT environment, *Expert Systems with Applications*, vol. 249, Sep. 2024, Art. no. 123080. <https://doi.org/10.1016/j.eswa.2024.123808>
- [3] M. Umer, M. Tahir, M. Sardaraz, M. Sharif, H. Elmannai, and A. D. Algarni, Network intrusion detection model using wrapper based feature selection and multi head attention transformers, *Scientific Reports*, vol. 15, Aug. 2025, Art. no. 28718. <https://doi.org/10.1038/s41598-025-11348-5>
- [4] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, and H. Wang, Retrieval-augmented generation for large language models: A survey, *arXiv preprint arXiv:2312.10997*, Dec. 2023.
- [5] S. Zhao, Y. Yang, Z. Wang, Z. He, L. K. Qiu, and L. Qiu, Retrieval augmented generation (RAG) and beyond: A comprehensive survey on how to make your LLMs use external data more wisely, *arXiv preprint arXiv:2409.14924*, Sep. 2024.
- [6] X. Li, Z. Zheng, M. Zhao, Y. Zhao, L. Shi, and B. Wang, RLFE-IDS: A framework of intrusion detection system based on retrieval augmented generation and large language Model, *Computer Networks*, vol. 268, Aug. 2025, Art. no. 111341. <https://doi.org/10.1016/j.comnet.2025.111341>
- [7] F. Blefari, C. Cosentino, F. A. Pironti, A. Furfaro, and F. Marozzo, CyberRAG: An agentic RAG cyber attack classification and reporting tool, *Future Generation Computer Systems*, vol. 176, Mar. 2026, Art. no. 108186. <https://doi.org/10.1016/j.future.2025.108186>
- [8] R. Devendiran and A. V. Turukmane, Dugat-LSTM: Deep learning based network intrusion detection system using chaotic optimization strategy, *Expert Systems with Applications*, vol. 245, Jul. 2024, Art. no. 123027. <https://doi.org/10.1016/j.eswa.2023.123027>
- [9] Z. Long, H. Yan, G. Shen, X. Zhang, H. He, and L. Cheng, A Transformer-based Network Intrusion Detection Approach for Cloud Security, *Journal of Cloud Computing*, vol. 13, no. 1, Jan. 2024, Art. no. 5. <https://doi.org/10.1186/s13677-023-00574-9>
- [10] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, Cyber threat intelligence sharing scheme based on federated learning for network intrusion detection, *Journal of Network and Systems Management*, vol. 31, iss. 1, p. 3, Oct. 2022. <https://doi.org/10.1007/s10922-022-09691-3>
- [11] M. Guarascio, N. Cassavia, F. S. Pisani, and G. Manco, Boosting cyber-threat intelligence via collaborative intrusion detection, *Future Generation Computer Systems*, vol. 135, pp. 30–43, 2022.
- [12] A. Lekssays, U. Shukla, H. T. Sencar, and M. R. Parvez, TechniqueRAG: Retrieval Augmented Generation for Adversarial Technique Annotation in Cyber Threat Intelligence Text, pp. 20913–20926, Jul. 2025. <https://doi.org/10.18653/v1/2025.findings-acl.1076>
- [13] L. Breiman, Random forests, *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001. <https://doi.org/10.1023/A:1010933404324>
- [14] M. Sakurada and T. Yairi, Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction, Dec. 2014. <https://doi.org/10.1145/2689746.2689747>
- [15] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection, Jul. 2016. <https://doi.org/10.48550/arXiv.1607.00148>
- [16] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, 1D convolutional neural networks and applications: A survey, *Mechanical Systems and Signal Processing*, vol. 151, Apr. 2021, Art. no. 107398. <https://doi.org/10.1016/j.ymssp.2020.107398>