



Enhancing federated intrusion detection through LLM-Driven alert enrichment and collaborative threat information sharing

Pablo Fernández Saura , Jorge Bernal Bernabé , Antonio Skármeta Gómez 

University of Murcia, C. Campus Universitario, 11, Murcia, 30100, Spain

ARTICLE INFO

Keywords:

Intrusion detection systems
Federated learning
Large language models
Retrieval augmented generation
Cyber threat intelligence
Non-IID data

ABSTRACT

The potential of Federated Learning (FL) to enhance Intrusion Detection Systems (IDS) is often compromised by performance degradation in non-independent and identically distributed (non-IID) data environments. While novel and powerful solutions have been proposed to mitigate this challenge, such as tailored aggregation algorithms, the near-perfect detection performance and low false positive rate needed for many critical use cases often remain elusive. This paper proposes a novel FL-based framework where participant organizations leverage a Large Language Model (LLM) to re-evaluate and enrich in real-time the low-confidence alerts produced by their base, FL-trained detection model. The LLM context for each alert is critically enhanced by two tiers of Threat Intelligence (TI): external TI (e.g., IP reputation services) and inter-organizational TI derived from a Federated Retrieval Augmented Generation (RAG) Incident Store of collaboratively shared, verified attack summaries. Furthermore, high-confidence LLM-verified alerts are fed back to continuously re-train and update the base detection model. Our experimental results, conducted on a challenging non-IID scenario using the CIC-IDS-2017 dataset where one organization completely lacked data from one attack type, demonstrate the framework's effectiveness. With full TI context, 2 out of the 4 analyzed LLMs achieved perfect recall in re-classifying low-confidence alerts from the missing class. The feedback loop using these LLM-verified samples boosted the base detection model recall from 75.85 % to 91.47 % (F1-score from 86.27 % to 95.54 %) on the missing attack class. Additionally, an approximate 1 % reduction in the False Positive Rate (FPR) was achieved, while inference times demonstrate potential for near-real-time applications.

1. Introduction

The collaborative training of Intrusion Detection Systems (IDS) using Federated Learning (FL) offers a powerful, privacy-preserving alternative to traditional centralized machine learning. By training models locally and only sharing parameter updates, organizations can build more robust detection models without exposing sensitive network data [1,2]. However, the practical effectiveness of FL is often compromised by a foundational challenge: the non-independent and identically distributed (non-IID) nature of real-world data [3,4]. As different organizations face different types of threats, their local datasets become skewed, which can lead to severe model performance degradation and slow convergence, undermining the very purpose of collaborative defense [3]. While advanced aggregation algorithms like FedProx and FedAdam have been proposed to mitigate this, they often fall short of achieving the near-perfect detection rates required for critical security applications. Furthermore, these approaches do little to address the inherent 'black box' nature of the resulting models, which limits trust and auditability.

This paper proposes a novel, FL-based framework that introduces a paradigm shift from simple model averaging to a more comprehensive "federated intelligence" system. The core of this architecture is a local Large Language Model (LLM) that acts as an intelligent secondary analysis engine, tasked with re-evaluating low-confidence alerts generated by the base FL-trained detection model in near-real-time. To ensure a robust and accurate re-evaluation, the LLM's context is critically enhanced by two tiers of Threat Intelligence (TI): (1) real-time external TI from public sources (e.g., IP reputation services), and (2) private, inter-organizational TI retrieved from a Federated Retrieval-Augmented Generation (RAG) Incident Store. This novel component allows participating organizations to collaboratively share semantic knowledge from verified incidents in a privacy-preserving manner. Finally, the framework closes the loop by using high-confidence, LLM-verified alerts to continuously retrain and improve the base detection model, while also providing human-readable explanations for its decisions.

The main contributions of this work are summarized as follows:

* Corresponding author.

E-mail addresses: pablofs@um.es (P. Fernández Saura), jorgebernal@um.es (J. Bernal Bernabé), skarmeta@um.es (A. Skármeta Gómez).

<https://doi.org/10.1016/j.future.2025.108319>

Received 28 June 2025; Received in revised form 29 October 2025; Accepted 11 December 2025

Available online 17 December 2025

0167-739X/© 2025 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

- A pioneering framework that synergistically combines Federated Learning (FL), Large Language Models (LLM), and Federated Retrieval Augmented Generation (RAG) to enable collaborative, context-aware intrusion detection.
- A novel architecture where an LLM enhances federated intrusion detection by leveraging a two-tiered Threat Intelligence system composed of external feeds and a collaborative RAG knowledge base.
- The application of privacy-preserving Federated RAG for sharing rich, semantic-based threat knowledge, moving beyond the limitations of traditional Indicator of Compromise exchanges.
- A closed-loop retraining mechanism where LLM-verified data demonstrably teaches a federated model to detect an attack class entirely missing from one participant's dataset.
- An experimental evaluation showing a recall increase on the missing class from 75.85% to 91.47% while reducing the overall False Positive Rate.

The remainder of this paper is organized as follows. [Section 2](#) reviews the state-of-the-art literature. [Section 3](#) details the design of the proposed framework. [Section 4](#) outlines the experimental evaluation methodology, and [Section 5](#) presents the evaluation results. Finally, [Section 6](#) discusses the challenges and limitations of the framework, and [Section 7](#) concludes the paper.

2. Background & related work

The framework proposed in this paper is multi-faceted, integrating concepts from several key research domains at the intersection of FL, LLMs, RAG, and collaborative cybersecurity. To properly contextualize our contribution, this section reviews the state-of-the-art in each of these areas. We first cover existing approaches to the foundational non-IID challenge in Federated Learning. Next, we review the diverse applications of LLMs in cybersecurity, followed by a closer look at RAG for context enrichment. Finally, we discuss the importance of Explainable AI in security analysis to frame the novel explanatory capabilities of our system.

2.1. The non-IID challenge in federated intrusion detection

The degradation of model performance in non-IID data settings is arguably the most significant barrier to the widespread, real-world adoption of FL for cybersecurity. The problem is so foundational that recent surveys highlight a lack of consensus in its very classification and quantification, creating detailed taxonomies of data heterogeneity [3]. Further relevant surveys categorize mitigation strategies, noting that while data-centric interventions exist, it is more common for researchers to propose improvements at the model, algorithm, or framework level [4]. Prominent algorithmic solutions focus on aggregation strategies tailored for non-IID data; for instance FedProx, which introduces a proximal term to stabilize local training on heterogeneous data [5], SCAFFOLD, which uses control variates to correct for client drift [6], or FedAdam that employs adaptive server-side optimization [7]. In contrast to these algorithmic modifications, data-centric approaches have also been explored. A notable example involves using generative models, particularly auxiliary classifier generative adversarial networks (ACGANs), to create synthetic data for specific minority or missing classes before training, aiming to rebalance local datasets to appear more uniform [8].

This work introduces a novel architectural solution distinct from established paradigms like altering core FL aggregation algorithms or relying on preemptive synthetic data generation. Instead, our approach focuses on reactively enhancing performance for specific underrepresented classes at inference time. This is achieved using an LLM, enriched with multi-source threat intelligence, to re-evaluate low-confidence alerts and a feedback loop to continuously retrain the base model with real, verified samples. We validated this approach under a formally defined "overlapping class allocation" scenario [3] to prove its effectiveness in a challenging, realistic non-IID setting.

2.2. LLMs applied to cybersecurity

The application of LLMs to cybersecurity is a rapidly maturing field, with comprehensive reviews documenting the trend of using LLMs to enhance existing security frameworks [9–11]. When it comes to performance, the potential of LLMs is clear: studies like [12] have demonstrated that LLMs can achieve near-perfect detection accuracy (>99.9%) on standard IDS datasets like CIC-IDS-2017 in ideal, centralized settings, providing a crucial performance benchmark. However, other research highlights that superior performance is not guaranteed. For instance, Nwafor et al. [13], in a direct comparison with traditional machine learning, found that simply fine-tuning an LLM for intrusion detection did not produce results comparable to established state-of-the-art ensemble methods.

This work builds upon and extends these concepts. While the benchmark set in [12] is impressive, our framework is designed to solve the much harder problem of robust detection in a federated, non-IID environment. Furthermore, while works such as [14] propose integrated detection and prevention pipelines and mention FL as a future direction, our framework already implements a sophisticated FL architecture as the basis of the system. The core novelty of our contribution lies not just in using an LLM for analysis, but in designing a complete, collaborative system around it where the reasoning is enriched by a fusion of external TI and a privacy-preserving, inter-organizational Federated RAG Incident Store. This context-rich architecture is what allows our LLM to succeed where simpler fine-tuning approaches may fail, directly addressing the gaps identified in the literature.

2.3. Explainability in AI-Driven security analysis

A significant barrier to the adoption of AI in high-stakes security operations is the "black box" nature of many machine learning models, which creates a trust deficit among security analysts. Research in Explainable AI (XAI) aims to solve this by making model decisions transparent. Traditional XAI approaches, such as LIME and SHAP, have been applied to IDS to provide feature-based explanations for model predictions [15,16]. However, a more recent paradigm positions LLMs themselves as powerful tools for XAI [17]. Instead of providing technical feature-importance scores, LLMs can offer "narrative explainability" by synthesizing complex data into a human-readable story. The framework proposed in this paper implements this advanced form of XAI. The LLM's "explanation of the rationale" moves beyond traditional XAI by not only considering the base model's inputs but also fusing them with the multi-source threat intelligence context to generate a coherent, intuitive, and actionable explanation valuable for both human security professionals and subsequent automated systems, such as mitigation managers or security orchestrators.

2.4. Retrieval augmented generation (RAG) for enhanced context in cybersecurity

A core component of our framework is RAG, a well-established technique that enhances LLMs by grounding them in rich, external, and up-to-date knowledge. In cybersecurity, this has been applied to strengthen analysis by integrating structured Knowledge Graphs [18] and to automate complex, specialized tasks like forensic timeline reconstruction [19], mapping threat reports to MITRE ATT&CK techniques [20], enforcing security policies [21] or providing real-time incident response guidance for SIEM alerts by retrieving context from security frameworks like NIST CSF [22]. Our framework's primary innovation is the application of Federated RAG, an emerging paradigm designed for secure, knowledge-intensive tasks in privacy-sensitive domains [23]. To the best of our knowledge, this work is the first to integrate this federated approach into a complete system that also leverages FL and LLMs for a collaborative cybersecurity application. The architecture in this paper

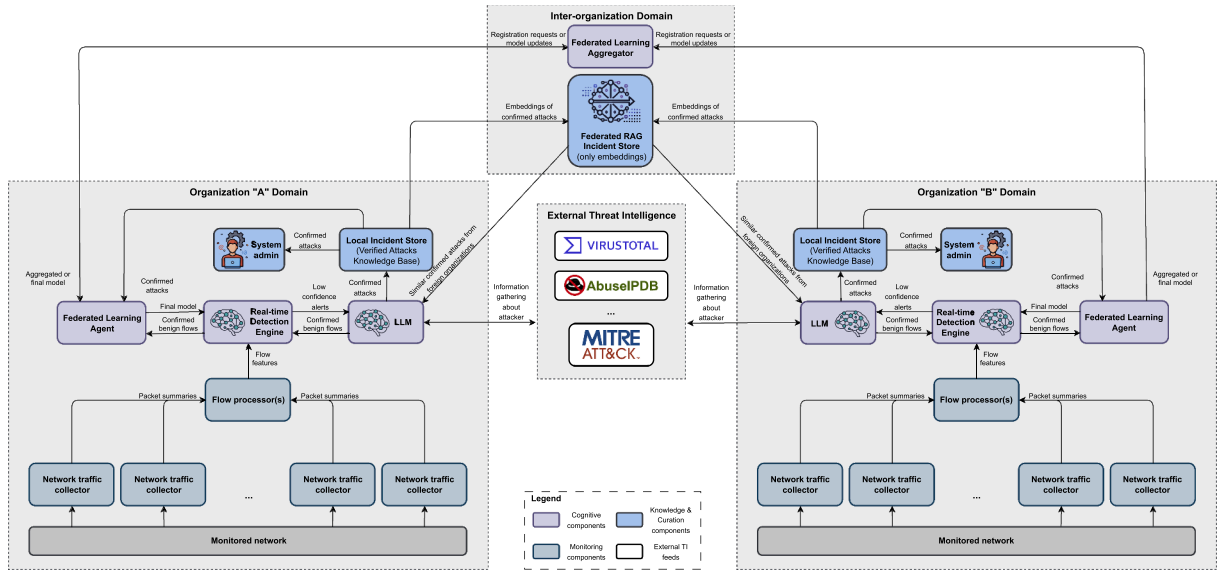


Fig. 1. Proposed framework's architecture.

extends this concept by combining two distinct TI sources: public, external feeds and a private, inter-organizational knowledge base to share confirmed incidents among organizations. Although powerful, the solution also introduces security challenges, most notably the risk of knowledge base poisoning, which is an active area of research, with studies proposing both general risk assessment frameworks [24] and specific forensic tools to trace such attacks [25].

3. Proposed framework

The architecture of the proposed solution is shown in Fig. 1. It is based on a distributed scenario involving an Inter-organization Domain and multiple federated organizations. While it depicts two such organizations for illustrative clarity, the system is designed to be scalable to support a larger federation. The framework's operation can be described in a stepwise manner, as detailed in the following subsections.

3.1. Network monitoring and feature extraction

Each organization features a set of traffic monitoring and processing tools that process and analyze traffic captured in real-time from the local network. Several Network Traffic Collectors perform Deep Packet Inspection (DPI) to extract relevant fields from the captured packets for the detection. Packet summaries containing this information are directed to one or several Flow Processors, which in turn collect all the information, create network flows and calculate a set of traffic statistics or features per flow. These flow features, which are the basis of the detection process, are used to feed the Real-time Detection Engine (see Fig. 1), which based on a detection model previously trained, classifies each received flow into benign or one specific attack class.

3.2. Federated training of the base model

Prior to the real-time monitoring and detection process, organizations have gone through the training phase. Each organization conforming the federation has a Local Incident Store, which contains data from attacks previously witnessed or learned by the organization in any other way (e.g., by integrating public intrusion detection datasets). This data is used to construct the training dataset for the local FL Agent. In addition to the malicious data, benign data samples are added to the training data continuously, coming from those predicted with high confidence by the base detection model, or those low-confidence ones verified by the

LLM. Once data is prepared, agents connect to the aggregator, which is hosted in the Inter-organization Domain, and a Federated Learning process starts. After several training rounds, in which specific model updates from agents have been created and aggregated, the final detection model is sent to the Real-time Detection Engine to perform inference.

3.3. Real-Time detection and alert triage

The Real-time Detection Engine, for each analyzed flow, creates a prediction report or alert that contains: (1) a timestamp representing when the flow was evaluated, (2) the source IP, (3) the destination IP, (4) the destination port, (5) the confidence of the prediction, and (6) the flow features. Alerts with a confidence below a pre-established threshold are sent to a local LLM which performs a dual role: it acts as both a secondary classifier to assign a final, context-informed attack label, and a reasoning engine to interpret the available evidence and generate an explanation.

3.4. LLM-driven alert enrichment and re-evaluation

For a more robust and informed response, the LLM's context is enhanced by two different sources of TI, as shown in the central part of Fig. 1:

1. **External TI sources:** to gather information about the source IP of the flow. Services such as VirusTotal and AbuseIPDB offer information about IP origin, reputation and historical data. In addition, other platforms such as MITRE ATT&CK can be consulted for more detailed information, e.g., specific mitigation techniques to be applied after the attack is confirmed.
2. **Inter-organizational TI:** to enable attack knowledge transfer between organizations in the federation, by using a Federated RAG Incident Store where organizations post embeddings of confirmed attacks from their Local Incident Store in a private manner, i.e., by sharing only a subset of the information and applying data anonymization techniques such as Differential Privacy to the sensitive parts. This way, when a local LLM is trying to evaluate a low-confidence alert, it can query, based on a similarity search, this federated database for similar incidents witnessed by the other organizations. This will be extremely helpful when other organizations have suffered an attack from the same source IP, or from a different source IP but a similar attack type.

The LLM will then re-evaluate the flow based on the information contained in the low-confidence alert (including the original flow features), augmented by context from the two TI sources. At the end of the process, it will generate a report containing the final attack label, the confidence level of the prediction, and an explanation of the rationale behind the decision.

3.5. Closed-Loop retraining and knowledge sharing

Only those flows surpassing another confidence threshold are considered as confirmed attacks. This additional filtering is used to minimize the risk of using incorrectly predicted samples in a new training process, which would poison the base detection model. These verified attacks are sent to the local system administrator or further automated systems for mitigation, pushed into the Local Incident Store, and consequently, embeddings are calculated, anonymized, and uploaded to the Federated RAG Incident Store to serve as context for other organizations. In addition, these are used to expand the training data, so the FL agent leverages them for further federated training processes of the base detection model.

This architecture not only enables an efficient, robust and communicative way of re-evaluating low-confidence alerts produced by the base model, which in turn serve to improve it at the same time, but also provide extensive explanations, provided by the LLMs, about the evaluation process and the rationale behind each decision taken. This explainability provides critical transparency for both system administrators and subsequent automated systems. By revealing not only the nature of network events but also the reasoning behind the model's conclusions, the framework empowers them to make more informed and effective strategic decisions.

4. Experimental evaluation

To assess the performance of the proposed framework, a non-IID FL scenario built upon the well-known CIC-IDS-2017 dataset [26] and composed of two organizations, i.e., Organization A and Organization B, was created. The selection of this specific dataset over other, more up-to-date alternatives is justified in Section 6.3.2.

sec:dataset-timeliness

Regarding the non-IID scenario set-up, we implemented this two-organization federation as a deliberate methodological choice, designed to create a clear, controlled, and interpretable environment. This allowed us to precisely measure the effectiveness of the core knowledge-sharing mechanism before scaling to more complex federations (discussed in Section 6.3.3). To create this specific setting, all samples from one specific attack type, i.e., Brute Force Attack (from now on, BFA), were completely removed from Organization A's dataset. The BFA class was strategically selected for this purpose. As a significant minority class with 13,835 samples, its removal creates a realistic knowledge deficit for Organization A. Yet, it is substantially more prevalent than extremely rare classes like Infiltration (36 samples), providing a sufficient number of test samples to robustly measure the model's ability to learn the missing class via the proposed framework.

The objective was to test how the framework enabled Organization A to learn the missing attack type by leveraging a locally deployed LLM, whose context was augmented by TI from Organization B and external sources. To this end, different LLMs were tested, and key performance metrics were extracted and analyzed. These metrics represented the effectiveness of each LLM in classifying low-confidence alerts, and the improvement in the base detection model's ability to predict the missing attack class after being retrained on LLM-verified data.

While a comprehensive evaluation of the base model involves multiple metrics, the performance for these specific tasks was assessed using the following standard metrics:

- Accuracy: $\frac{TP+TN}{TP+FP+FN+TN}$

- Recall: $\frac{TP}{TP+FN}$
- F1-Score: $2 * \frac{\text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}$

where TP: true positives, TN: true negatives, FP: false positives, and FN: false negatives.

For the specific task of evaluating the LLM's output, Recall was used as the primary metric. This choice is driven by the nature of the task, which focuses on identifying attacks that the base model missed. Therefore, this metric provides the most direct measure of the LLM's effectiveness in identifying these previously undetected threats. Similarly, Recall on the BFA class was also the main performance indicator for the base detection engine, as the central goal of the experiment is to measure its ability to learn and correctly identify this missing class. The full suite of metrics, including Accuracy and F1-Score, was employed for the broader evaluation of the base model's overall performance across all traffic types.

The specific LLMs chosen for this evaluation, listed below, were selected based on a focus on open-source candidates in the approximately 8 billion parameter range, as this was identified to offer a strong trade-off between accuracy and inference time for this real-time use case. The selection also included a model, the Zysec-7B, specifically fine-tuned for cybersecurity to assess the benefits of domain specialization.

1. Zysec-7B - 7.24B parameters [27]
2. Llama-3.1-8B-Instruct - 8.03B parameters [28]
3. Ministral-8B-Instruct-2410 - 8.02B parameters [29]
4. Gemma-2-9B-Instruct - 9.24B parameters [30]

This section is divided into four main sub-sections. The first describes the data pre-processing phase, where the original dataset is processed and transformed to meet the requirements needed for the designed non-IID scenario. The second details the construction of the baseline, which represents the best possible scenario without the LLM-based enrichment pipeline, i.e., a standard FL implementation with optimal parameters and aggregation functions, which will be used to compare and assess the effectiveness of the proposed solution. The third reports the implementation and tested features. Lastly, the fourth outlines how the different experiments have been created and executed.

4.1. Preparing the data

The data preprocessing phase started with the original dataset in tabular format. In its raw form, the dataset was composed of 3,119,345 samples or traffic flows and 85 columns or flow features. Initially, there were 14 attack types plus benign samples. The attack types included different kinds of DoS and DDoS attacks, brute force attacks based on FTP, SSH and web protocols, port scans, bot attacks, infiltration and Heartbleed exploit.

First, a data cleaning process was performed, where infinite and null values were imputed. For this, the mean of the column was used for numeric features, while the mode was used for categorical features. Additionally, features with constant values were removed. In total, 24,536,904 infinite/null values were imputed and 8 constant features were removed.

Next, to address multicollinearity within the dataset, highly correlated features were identified and one feature from each highly correlated pair was removed. To do this, the correlation matrix was calculated and pairs exceeding a conservative threshold of 0.9 were targeted for removal. This high threshold is a standard practice chosen to eliminate only features with near-perfect redundancy, which can prevent model instability, while avoiding the premature removal of features that might contain unique, valuable information at more moderate correlation levels. To further validate this choice, a sensitivity check was performed. The analysis confirmed the robustness of our approach by revealing that the highest correlation value below our 0.9 threshold was 0.8954. This means the exact same set of features would have been removed using

any threshold between 0.8955 and 0.9. This process resulted in the removal of a total of 31 columns.

For the sake of simplicity, attack labels were grouped into higher-level types, e.g., DoS GoldenEye to DoS Attack or FTP-Patator into Brute Force Attack (BFA). The number of labels was reduced from the initial 15 to 9.

Finally, numeric features were standardized by removing the mean and scaling to unit variance. 10% of the samples were reserved for testing, while the remaining 90% were equally divided and used to create the two datasets assigned to each organization. This standard partition ratio was employed to ensure the creation of a robust baseline model for our experimental validation; it does not represent a real-world data availability requirement. The class distribution was preserved during this process, so that each organization had a similar number of samples for each attack type. Then, all BFA samples were removed from Organization A's dataset, to create the non-IID setting. Lastly, identifier features, such as the flow identifier, source and destination IP and ports, and timestamp, were removed for training. With respect to the solution's architecture, this would be equivalent to initially populating the Local Incident Store with a publicly available intrusion detection dataset.

4.2. Constructing the baseline

In order to establish a representative baseline against which to measure the framework's effectiveness, an exhaustive comparison of different FL aggregation strategies was carried out. In particular, the following aggregation algorithms were considered: FedAvg [1], FedProx [5], FedAdam [7], and SCAFFOLD [6]. For FedProx, its proximal term (μ), for FedAdam, its server-side learning rate (η), and for SCAFFOLD, its client-side learning rate (η_c), were fine-tuned and compared. Apart from FedAvg, the rest of the algorithms are specifically designed to work in non-IID environments like ours.

The FL process, composed of 2 FL Agents (one per organization), was configured to run 20 training rounds, with 5 local epochs, a learning rate of 0.01 and a batch size of 256. Organization A's training dataset consisted of 1,397,479 samples, while Organization B's had 1,403,705, with the difference resulting from the removal of all BFA samples from Organization A's data. The testing data portion, used to extract the final performance metrics, consisted of 311,935 samples, with all 9 attack types represented.

The comparison points chosen were: (i) average rounds to achieve 90% in accuracy, (ii) average final accuracy, and (iii) average final recall and F1-score on BFA samples from testing data.

For the base detection model, a preliminary evaluation was conducted to select the most suitable architecture. Several algorithms commonly used for intrusion detection were tested, including Random Forest (RF) and Support Vector Machines (SVM). A Multi-layer Perceptron (MLP) neural network was ultimately selected as it demonstrated superior performance in our experimental setting. The chosen MLP was composed of an input layer with a neuron per feature, two hidden layers of 64 neurons each, and the output layer with a neuron per attack type. ReLU was used as activation function. This model was trained by FL and deployed in the Real-time Detection Engine, where performance metrics were calculated based on the testing data.

With this setting, 5 seeded runs were executed per aggregation algorithm and hyperparameter setting, if applicable. The model from the best-performing run (primarily evaluated on the BFA class recall) was selected for installation in the Real-time Detection Engine, where it was used to carry out the following experiments.

4.3. Implementation and testbed details

The experiments were conducted on a server equipped with an AMD EPYC 9554 64-Core CPU, 1 TB of RAM, and two NVIDIA H100 NVL (94 GB each) GPUs. The software environment was built on Debian 12 with Python 3.10. The FL components and the detection model were

implemented using Flower v1.18 and PyTorch v2.6, respectively. The Federated RAG Incident Store was deployed using OpenSearch v2.15. Lastly, the LLMs were served using CUDA v12.1, vLLM v0.8.5.post1 and FlashInfer v0.2.2.post1, with xGrammar as guided decoding backend to enforce the format of responses. Each LLM was deployed on a single GPU for all experiments; no model or tensor parallelism was utilized across the two available GPUs.

4.4. Preparing the experiments

Once the baseline was set up and the detection model was running inside the Real-time Detection Engine, the experiments described in the following subsections were carried out. For all of them, the Organization A domain was considered.

4.4.1. Brute force attack on-Inference evaluation and training data feedback

For this experiment, the BFA samples were isolated from the testing data. The attacker's IP was replaced in every sample by a known malicious IP from AbuseIPDB. Since there was only one source IP for all BFA flows in the testing data, only one real IP was used. The base detection model was fed with these samples, simulating a real-time attack being monitored. The model generated one report or alert for each evaluated traffic flow, with the format described in Section 3.3. Since a confidence value was included in each report, which indicates the model's confidence in its prediction, a confidence threshold was set to 85% and output alerts were filtered, so that only those with a confidence below the threshold would advance through the pipeline. The selection of that specific threshold value was justified by a density distribution analysis and an ablation study described in Section 5.2. Then, these resulting low-confidence alerts were sent to the local LLM, which re-evaluated them and provided the final attack label.

It is important to note that while the architecture (see Fig. 1) is designed to integrate with various external TI sources, this evaluation focused specifically on real-time IP reputation as the primary form of external context. For this purpose, AbuseIPDB was selected as the sole TI provider. This decision was based on preliminary analysis indicating that, for the specific malicious IP address used in our testbed, AbuseIPDB provided the most consistent and unambiguously negative reputation score. While other excellent services like VirusTotal offer similar data, AbuseIPDB's strong signal was chosen to ensure the LLM received the most deterministic context possible for this controlled experiment. Other valuable sources like MITRE ATT&CK, although relevant at solution design level, were considered out of scope for this specific re-classification task, as its focus is on higher-level strategic intelligence, such as adversary Tactics, Techniques, and Procedures (TTPs) and mitigation strategies, rather than the immediate reputation of a potentially malicious IP address.

The Federated RAG Incident Store, implemented with OpenSearch, was initially populated with attack samples, i.e., 1000 randomly selected for each attack class, from each organization's training data contained in the Local Incident Store. It should be noted that this population method represents a straightforward approach chosen for experimental tractability. While sufficient to validate the knowledge-sharing concept in our tests, a practical, large-scale deployment would necessitate future research into more sophisticated sampling strategies (e.g., based on incident severity or temporal relevance) to ensure the knowledge base is both compact and maximally informative.

The uploaded information included attacker's IP, victim's destination port, attack type, transport-layer protocol used, timestamp when the incident occurred, source organization, and the first five flow features. To ensure data privacy, Local Differential Privacy based on Laplacian noise adding was applied over the shared flow features, using a strong privacy factor of $\epsilon = 1$. The embeddings were generated using the well-known sentence transformer model *all-MiniLM-L6-v2*. This model was

specifically chosen for its high efficiency and strong performance in generating quality semantic embeddings, providing an excellent balance between speed and accuracy that is critical for near-real-time applications. For each alert to be evaluated by the LLM, the 5 most similar incidents were retrieved based on a k-Nearest Neighbor (k-NN) semantic search. The value of $k = 5$ was determined through preliminary experiments to offer the optimal trade-off between providing sufficient context for the LLM and avoiding excessive noise from less relevant incidents, which could increase processing load without improving classification accuracy.

The considered LLMs performance was tested under the following context scenarios: (i) plain, where only the alert information was provided, (ii) plus AbuseIPDB TI, where a report for the flow's source IP was obtained in real-time from AbuseIPDB and added in addition to the plain alert as context, and (iii) plus inter-organizational RAG, where the Federated RAG Incident Store, previously populated, was queried in real-time and the 5 most similar incidents to the current alert were retrieved and added to the plain alert and the AbuseIPDB TI information. For each evaluated alert, the source IP was used to query the AbuseIPDB service in real-time. Since attacker's IP was replaced by a real malicious IP in each sample, the response from the service returned an actual report indicating malicious behavior, i.e., negative confidence score, high number of reports, etc.

The output re-evaluated alerts from the best performing model were again filtered by another confidence threshold, set at 90 %. This high, conservative threshold represents a deliberate safety trade-off: we are prioritizing the long-term integrity of the base model by minimizing poisoning risk, even at the cost of potentially discarding some correctly-classified samples that fall below 90 % confidence. While this is a sound practice for a safety filter, we also acknowledge that this uniform threshold is a methodological limitation for direct comparison, as different models have distinct confidence calibration characteristics (i.e., a 90 % score from *Gemma-2-9B-Instruct* may not be equivalent to 90 % from *Llama-3.1-8B-Instruct*). Alerts re-evaluated with a confidence greater than this second threshold were re-inserted to the Local Incident Store, and thus added to the Organization A's training data. At this point, a new FL training process was initiated with Organization B's FL Agent, and the base detection model's recall on BFA class was measured and compared to the baseline.

4.4.2. Detection model false positives rate reduction

For this experiment, all benign samples were extracted from the testing data. Similarly to the previous experiment, these samples were fed to the base detection model, reports/alerts were generated, filtered by confidence, and sent to the LLMs. This experiment demonstrated the LLMs' effectiveness in also reducing the False Positive Rate (FPR) of the base model.

In this case, since benign samples are not included in Local Incident Store, and thus not present in the Federated RAG Incident Store, the RAG process for this experiment returned no hits for any evaluated alert. Similarly, queries to the AbuseIPDB service returned reports indicating non-malicious behavior as reported IPs were not included in the platform database.

4.4.3. Inference time analysis

This experiment featured an exhaustive evaluation of LLMs per-alert inference times for the three different TI scenarios. In addition, the experiment measured how varying the number of concurrent requests being sent to the LLM affected the inference time, simulating different batch sizes or load spikes that may occur in a real-time environment. For this time analysis, the Brute Force Attack On-Inference Evaluation and Training Data Feedback experiment scenario (see [Section 4.4.1](#)) was used. Each setting was run 3 times and the results were averaged.

4.4.4. Ablation study: Comparison with a traditional classifier

To address whether a computationally expensive LLM is necessary for the alert re-evaluation task, a comparative ablation study was conducted against two well-known machine learning models: eXtreme Gradient Boosting (XGBoost) and Random Forest (RF). These were chosen for their consistent high performance on structured data and their common use as strong industry baselines. We sought to determine if a traditional model could achieve similar results if provided with the same enriched contextual information.

To this purpose, we generated a dataset by gathering all the low-confidence alerts produced by the base model from the original test set, resulting in 17,180 samples comprising benign and all the attack labels. For a direct comparison, in addition to the original numerical flow features list, we added a set of features to represent both the TI information coming from AbuseIPDB service and the Federated RAG Incident Store.

The resulting enriched dataset was split into training (66 %) and testing (33 %) sets. The models were then trained for a multi-class classification task to predict the ground-truth label of each alert. To ensure a fair baseline comparison without extensive hyperparameter tuning, both models were built using their library-default settings (e.g., 100 estimators for the RF), with a fixed random seed to ensure reproducibility. The primary evaluation metric was again the model's recall in correctly identifying the misclassified BFA samples from the test set, allowing for a direct comparison with the LLM's performance on the same task.

4.4.5. Robustness evaluation under ambiguous external threat intelligence

Lastly, to address the potential limitation of relying on a single, known malicious IP in the initial evaluation, a further experiment was conducted to assess the framework's robustness under more realistic conditions with varying external TI quality. Using the same 342 low-confidence BFA alerts identified previously, we simulated three distinct scenarios by modifying the external AbuseIPDB information provided to the LLMs:

1. **No Record:** Simulating an attack from a new IP address not yet reported (AbuseIPDB score and report count set to 0).
2. **Low/Suspicious:** Simulating an ambiguous signal where an IP has minimal negative history (low score and report count).
3. **Known Bad:** Replicating a strong signal where the IP is clearly malicious (high score and report count).

The original 342 alerts were split in three equal-size sets, each one mapping to one of the previously described scenarios. All four LLMs were then re-evaluated on these alerts, allowing for an analysis of their performance when external TI is absent, weak, or contradictory to the internal RAG context.

5. Evaluation results

5.1. Baseline

As stated in [Section 4.2](#), to achieve the best standalone FL setting, four different aggregation algorithms were evaluated, namely: FedAvg, FedProx, FedAdam, and SCAFFOLD.

Results are shown in [Table 1](#), where each column represents each of the listed comparison points, and each row represents the different aggregation algorithms with different main hyperparameter values (not applicable to FedAvg). Since 5 different seeded runs were executed and averaged, the standard deviation across the executed runs is also included for every column. Although a wider range of hyperparameter values for FedProx and FedAdam were tested, the table includes only those with significant results for simplicity.

As can be concluded from the results, FedAdam with $\eta = 0.08$ stands out as the best configuration to predict the BFA samples, achieving a mean of 64.53 % recall and 78.07 % F1-score, outperforming the other options regarding these two comparison points. The convergence

Table 1

Different Federated Learning strategies performance on the non-IID scenario.

Strategy	Avg Rounds to >90 % Acc ($\pm$ std)	Avg Final Acc (Overall) ($\pm$ std)	Avg Final Recall (Brute Force) ($\pm$ std)	Avg Final F1 (Brute Force) ($\pm$ std)
FedAvg	1.0 $\pm$ 0.0	0.9740 $\pm$ 0.0015	0.4952 $\pm$ 0.0146	0.6622 $\pm$ 0.0130
FedProx ($\mu = 0.01$)	1.0 $\pm$ 0.0	0.9739 $\pm$ 0.0007	0.3602 $\pm$ 0.0941	0.5223 $\pm$ 0.1064
FedAdam ($\eta = 0.03$)	5.2 $\pm$ 2.5	0.9490 $\pm$ 0.0055	0.5403 $\pm$ 0.1256	0.6934 $\pm$ 0.1000
FedAdam ($\eta = 0.05$)	3.6 $\pm$ 1.9	0.9616 $\pm$ 0.0080	0.3261 $\pm$ 0.1962	0.4559 $\pm$ 0.2430
FedAdam ($\eta = 0.08$)	2.8 $\pm$ 1.2	0.9552 $\pm$ 0.0069	0.6453 $\pm$ 0.0902	0.7807 $\pm$ 0.0677
SCAFFOLD ($\eta_c = 0.03$)	1.0 $\pm$ 0.0	0.9878 $\pm$ 0.0009	0.5053 $\pm$ 0.1204	0.6624 $\pm$ 0.1126
SCAFFOLD ($\eta_c = 0.05$)	1.0 $\pm$ 0.0	0.9872 $\pm$ 0.0013	0.5866 $\pm$ 0.0733	0.7369 $\pm$ 0.0549
SCAFFOLD ($\eta_c = 0.1$)	1.0 $\pm$ 0.0	0.9827 $\pm$ 0.0120	0.5416 $\pm$ 0.1044	0.6967 $\pm$ 0.0877

speed, represented in the first column, was significantly slower compared to FedAvg, FedProx or SCAFFOLD, but faster if compared to the other FedAdam configurations. Additionally, the mean final accuracy across all classes (second column) was notably lower compared to the other high-performing strategies, such as SCAFFOLD ($\eta_c = 0.05$) which achieved 0.9872. This indicates that this aggregation algorithm aims to improve performance on underrepresented, e.g., BFA class which was completely missing from one organization's data, at the expense of some performance on the rest of the attack types, where the other analyzed algorithms achieve significantly better results.

Given that the model's performance on the missing class is the primary criterion for this baseline, the FedAdam configuration with $\eta = 0.08$ was selected as the best overall strategy. While Table 1 shows the average performance across all five runs, we chose the single best-performing model instance from these runs to serve as the strongest possible baseline for our subsequent experiments. This peak performance was achieved with seed 4, yielding a recall of 75.85 % and an F1-score of 86.27 % on the BFA test samples. This specific model instance was therefore installed in the Real-time Detection Engine to be used for the next experiments, and established as the baseline against which the enhancements of our proposed solution were measured.

5.2. Brute force attack on-Inference evaluation and training data feedback

Once the best standalone FL setting was found and set as the baseline to be improved, this experiment focused on evaluating the different LLMs under the conditions described in Section 4.4.1. After the detection model was fed with the BFA test samples (1,383 in total), a report or alert was generated for each one. These alerts were filtered by the confidence value provided by the model, using a threshold of 85 %. This threshold was selected based on a data-driven analysis composed of two parts.

First, the visual analysis in Fig. 2 shows the density distribution of the analyzed alerts, distinguishing between correct and incorrect classifications. It can be noted that, in the case of correctly predicted samples, the majority of them exceeded the threshold and grouped very close to the 100 % confidence value, indicating that the model was highly confident when making a correct prediction. Conversely, for incorrect predictions, most of the corresponding alerts' confidence values were condensed below the threshold. This showed that the selected threshold was in a near optimal region to achieve the objective, which was to catch those incorrectly predicted samples based on the prediction confidence and direct them through the re-evaluation pipeline.

Second, to provide a quantitative justification for this specific value, a sensitivity analysis was conducted. Table 2 presents the results, showing the trade-off between the percentage of BFA False Negatives (FNs) captured for re-evaluation (from a total of 334 present in the test set), and the total number of low-confidence alerts sent to the LLM at various thresholds, containing also false negatives from other attack classes. Results confirm the 85 % confidence threshold as an optimal balance for this experiment. It successfully captures a substantial majority (71.6 %) of the missed BFA samples while avoiding the much larger workloads associated with higher thresholds.

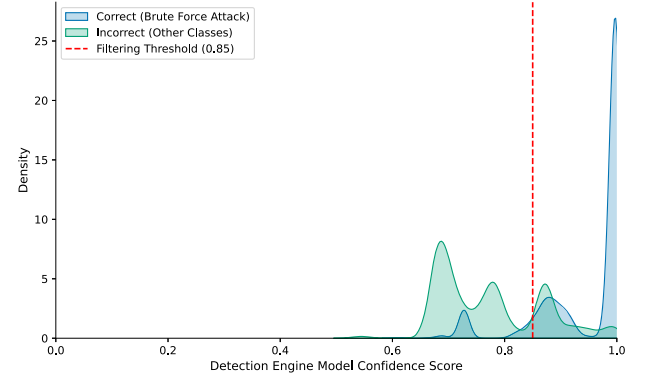


Fig. 2. Confidence distribution of Detection Engine model tested on Brute Force Attack samples.

In total, after this first confidence filtering, 342 BFA samples were selected out of the initial 1,383. These were used to feed each considered LLM and evaluate how they performed for each context setting (see Section 4.4.1). In Fig. 3, the Recall of each LLM on the BFA samples for each setting is shown. Since LLMs also include a prediction confidence value, the second confidence threshold (set at 90 %) was applied, and results shown in the figure represented samples re-evaluated with a confidence equal or greater to this confidence value.

As expected, for all LLMs, including more context information had a direct impact on performance, and the full context option (Plain alert + AbuseIPDB TI + RAG) was the one that achieved the best results overall, matching the expectations. In general, models tended to give high credit to the TI information coming through the RAG, as it confirms similar attacks witnessed by another organization in the federation, which is key. Comparing the models, *Ministral-8B-Instruct* and *Gemma-2-9B-Instruct* achieved perfect recall on the full context setting, while *Llama-3.1-8B-Instruct* and *Zysec-7B* reached 98 % and 76 %, respectively. It is worth noting that the *Zysec-7B* model, which is the only one specifically fine-tuned for cybersecurity, was the worst-performing one when given full context. This suggested that the model's general capabilities were more impactful than its domain specialization for this task. Notably, the performance issues with *Zysec-7B* extended beyond accuracy, as it was the only model that consistently failed to adhere to the required output format. Approximately 5 % of its responses were malformed and thus excluded from the evaluation—a failure that occurred even while using a format-enforcing backend. This is a significant practical drawback, as such failures would cause operational failures in a real-world, automated system. However, this model outperformed *Llama-3.1-8B-Instruct* in plain and AbuseIPDB TI settings, and *Gemma-2-9B-Instruct* in plain setting, possibly indicating a better ability to analyze the raw flow features, which is the only significant information provided in the plain option. *Gemma-2-9B-Instruct* model fails completely when no TI context is provided, but achieved almost perfect recall when AbuseIPDB information is given, pointing out that the source IP is very likely an attacker. This is a surprising finding, since a less pronounced improvement would be expected when adding the AbuseIPDB TI context, as happened

Table 2
Sensitivity analysis of the base model's confidence threshold.

Confidence Threshold (<)	BFA FNs Captured (%)	Total LLM Workload (Alerts)
0.70	32.9 % (110/334)	117
0.75	47.6 % (159/334)	227
0.80	68.9 % (230/334)	298
0.85	71.6 % (239/334)	342
0.90	90.1 % (301/334)	562
0.95	95.5 % (319/334)	646

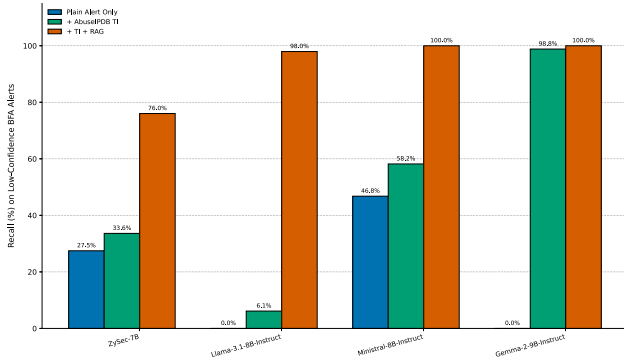


Fig. 3. LLMs recall in classifying BFA low-confidence alerts.

with *Llama-3.1-8B-Instruct*. Overall, *Ministral-8B-Instruct* model seemed the best for this experiment, achieving decent recall values with the most restrictive setting, and showing significant improvement when the different levels of TI are provided, reaching perfect recall on full context setting.

To provide a qualitative and concrete illustration of this re-evaluation pipeline, Fig. 4 depicts the end-to-end process using the full context setting for an example low-confidence BFA alert. The process begins with a flow that the base model erroneously classifies as benign with a low confidence of 68.02 %, triggering the re-evaluation pipeline. For presentation purposes, the source IP in the figure is anonymized, as it is the real malicious IP from AbuseIPDB used for replacement in BFA testing samples.

The figure then illustrates how the LLM synthesizes the two streams of intelligence. The external TI context confirms the source IP's malicious profile via AbuseIPDB, which shows a maximum abuse confidence score. Simultaneously, the inter-organizational TI, retrieved from the Federated RAG Incident Store through a semantic search, corroborates this by finding multiple similar BFA incidents reported by Organization B.

Finally, the LLM's assessment demonstrates the framework's power. By fusing these inputs, it corrects the classification to BFA and boosts the confidence to a decisive 95 %. The generated narrative explanation explicitly confirms this synthesis, referencing the original flow's features, the external TI report, and the shared similar incidents to justify its reasoning. As this re-evaluated alert surpasses the second confidence filtering threshold, the associated traffic flow is fed back into the loop to improve the base model in the next training process.

Considering the best performing setting, which could be either *Ministral-8B-Instruct* or *Gemma-2-9B-Instruct* with full context, the 342 BFA samples used as input were correctly assigned the ground-truth attack type. Although a further confidence-based filtering was applied to the LLMs output, only *Llama-3.1-8B-Instruct* did not assign a confidence value greater than the threshold for all considered samples. Therefore, the 342 samples were considered candidates to be re-inserted to Organization A's Local Incident Store and thus be used as new training data to improve the base detection model through a new FL training process which also leveraged the original training data. Table 3 shows the improvement of the base detection model with respect to the initial

baseline results. Columns in the table represent the initial average recall on BFA testing samples (non-overlapping with the new inserted ones), the recall after including the 342 re-evaluated samples, and the absolute and relative improvement. As can be clearly seen, recall was greatly improved for all aggregation algorithms except for FedProx, which was an unexpected result, likely attributable to FedProx's core proximal term. This mechanism, designed to prevent local model drift by penalizing large updates, may have overly constrained the significant changes required for Organization A to learn the new samples from the previously missing class, effectively treating valuable new knowledge as undesirable deviation and suppressing its integration into the global model. As for the rest, even though the best improvement and mean overall metrics were found when SCAFFOLD with $\eta_c = 0.05$ was used, the best seeded run was still achieved by the FedAdam with $\eta = 0.08$ configuration, which was the one previously established as the baseline. While a moderate average relative improvement of 7.7 % was achieved, the best seeded run yielded a 91.47 % recall and 95.54 % F1-score on the missing class, representing an impactful relative improvement of 20.59 % and 10.74 % in recall and F1-score, and beating the second-best performer (SCAFFOLD with $\eta_c = 0.05$) who achieved a 85.03 % recall on the missing class in its best seeded run.

5.3. Detection model false positives rate reduction

This experiment aimed to assess how the LLMs could reduce the number of false positives generated by the baseline model. To do this, the exact same pipeline used for the previous experiment, i.e., the one proposed in the solution architecture, was used. In this case, instead of BFA samples, benign samples were selected from testing data. In total, 256,170 samples were used.

First, these samples were predicted by the detection model. The model correctly identified 240,108 samples (93.73 %) as benign but misclassified 16,062 (6.27 %) as attacks. These were considered the initial set of false positives. As shown in Fig. 5, the confidence distribution for correct and incorrect predictions is depicted along with the 85 % confidence threshold. Again, the model was very confident when correctly identifying the samples, while confidence for the misclassified ones was lower. Consequently, a significant portion of the false positives (15,528 alerts) fell below the threshold and were directed to the LLM re-evaluation pipeline.

In Fig. 6, the different LLMs effectiveness (recall) on those low-confidence benign samples is shown. In contrast to the previous experiment where BFA samples were considered, values are significantly lower. This is mainly caused by the confidence given by all the LLMs in the predictions. Indeed, the LLMs are consistently less confident when predicting benign samples, so that the second confidence filtering discards a significant portion of the analyzed alerts. This is especially critical for *Gemma-2-9B-Instruct*, whose performance is seriously degraded because of this. However, as argued in Section 3, this additional filtering on the LLM outputs is implemented to minimize the risk of poisoning the detection model, and it is a good methodological practice to maintain the same threshold value regardless of the experiment (90 %). The rest of the models showed a decent performance. In general, the trend of an improvement in the recall when adding more levels of context was clear and similar to the BFA experiment. However, in the case of

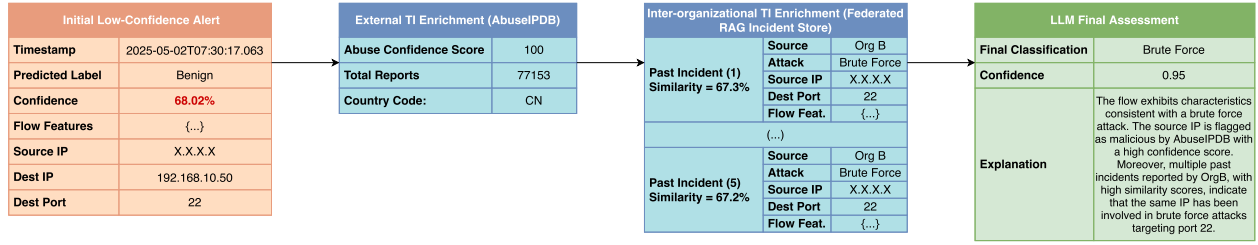


Fig. 4. Qualitative example of the end-to-end LLM-driven re-evaluation pipeline (using *Gemma-2-9B-Instruct*) for a single flow misclassified with low confidence by the base detection model.

Table 3

Recall improvement with confirmed low-confidence brute force attack alerts added to the training data.

Strategy	Initial BFA Recall ($\pm$ std)	After LLM BFA Recall ($\pm$ std)	Absolute Improve (pp)	Relative Improve (%)
FedAvg	0.4952 $\pm$ 0.0146	0.5621 $\pm$ 0.0071	+ 6.69 pp	+ 13.5 %
FedProx ($\mu = 0.01$)	0.3602 $\pm$ 0.0941	0.3297 $\pm$ 0.1526	-3.05 pp	-8.5 %
FedAdam ($\eta = 0.03$)	0.5403 $\pm$ 0.1256	0.5834 $\pm$ 0.0921	+ 4.31 pp	+ 8.0 %
FedAdam ($\eta = 0.05$)	0.3261 $\pm$ 0.1962	0.3608 $\pm$ 0.2266	+ 3.47 pp	+ 10.6 %
FedAdam ($\eta = 0.08$)	0.6453 $\pm$ 0.0902	0.6952 $\pm$ 0.2223	+ 4.99 pp	+ 7.7 %
SCAFFOLD ($\eta_k = 0.03$)	0.5053 $\pm$ 0.1204	0.7572 $\pm$ 0.0125	+ 25.19 pp	+ 49.85 %
SCAFFOLD ($\eta_k = 0.05$)	0.5866 $\pm$ 0.0733	0.7806 $\pm$ 0.0350	+ 19.4 pp	+ 33.07 %
SCAFFOLD ($\eta_k = 0.1$)	0.5416 $\pm$ 0.1044	0.7503 $\pm$ 0.0651	+ 20.87 pp	+ 38.53 %

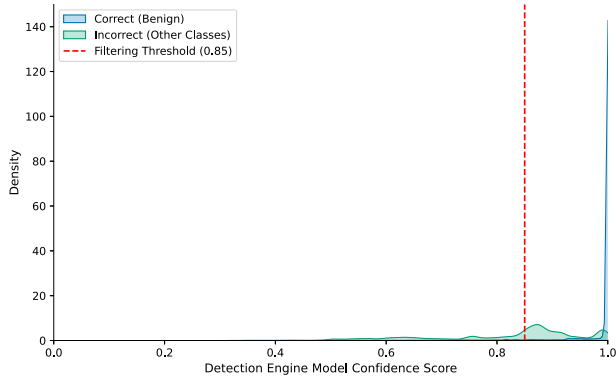


Fig. 5. Confidence distribution of Detection Engine model tested on benign samples.

Zysec-7B, the configuration where only the AbuseIPDB context about the flow's source IP is given in addition to the plain alert, stands out as the best-performing configuration overall. This is an unexpected behavior, since this LLM was the worst performing one in the previous experiment, and it is not logical that adding the inter-organizational RAG (indicating no similar incidents have been witnessed by Organization B) caused such a recall drop (from 72.5 % to 34.7 %). This result likely stems from a "security-first" bias in the specialized *Zysec-7B* model, highlighting a key fine-tuning trade-off. Unlike general-purpose models, which would interpret a "no hits" RAG result as evidence of benign behavior, the fine-tuned model might perceive this lack of a known signature as an indicator of a novel, zero-day threat. The model would then misclassify the flows as malicious, thereby degrading the recall. This suggests its specialization, while potentially useful for some tasks, comes at the cost of flexibility in reasoning about ambiguous, non-threatening data. *Llama-3.1-8B-Instruct* and *Ministral-8B-Instruct* achieved close performance, and outperformed *Zysec-7B* in the full context setting, while benefiting from each layer of TI context added. Again, *Ministral-8B-Instruct* stood out as the most consistent high-performer (aside from the anomalous peak achieved by *Zysec-7B* with the AbuseIPDB TI context configuration), confirming the good performance of this model within the proposed framework.

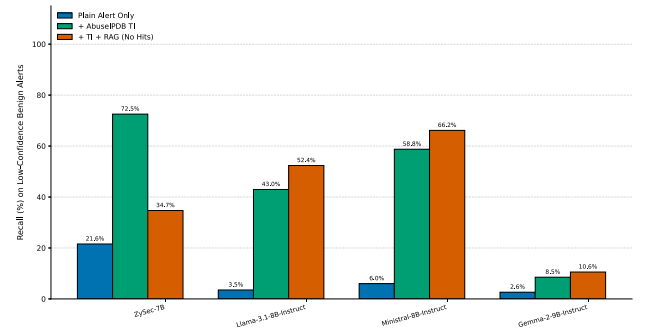


Fig. 6. LLMs recall in classifying benign low-confidence alerts for each context configuration.

The end-to-end impact on the FPR is significant. Initially, the base-line model alone produced 16,062 false positives (a 6.27 % FPR). Within our framework, the first confidence filter allowed most correct predictions to pass without LLM review, but it still left 9554 high-confidence false positives that were never re-evaluated. The crucial step was the LLM re-evaluation of the 15,528 low-confidence alerts. Here, the top configurations of *Zysec-7B* (with AbuseIPDB TI) and *Ministral-8B-Instruct* (with full context) failed to correct 4263 and 5256 alerts, respectively. Combining the original false positives that exceeded the first 85 % confidence threshold with those resulting from the LLMs re-evaluation, the total false positives for the full pipeline were reduced to 13,817 (*Zysec-7B* with AbuseIPDB TI) and 14,810 (*Ministral-8B-Instruct* with full context). This corresponds to a final FPR of 5.39 % and 5.78 %, representing a net reduction of 0.88 and 0.49 percentage points (pp), respectively. This result demonstrates that the framework contributes effectively to reducing the overall FPR. While not a drastic reduction, it is important to note that the confidence thresholds were kept fixed for methodological reasons and were not specifically optimized for this FPR reduction task.

However, this modest FPR reduction, when viewed in isolation, may not by itself justify the computational overhead of the LLM pipeline. The framework's primary value proposition relies on the effectiveness in solving the critical non-IID challenge, as demonstrated by the significant recall improvement on the missing BFA class (see Section 5.2).

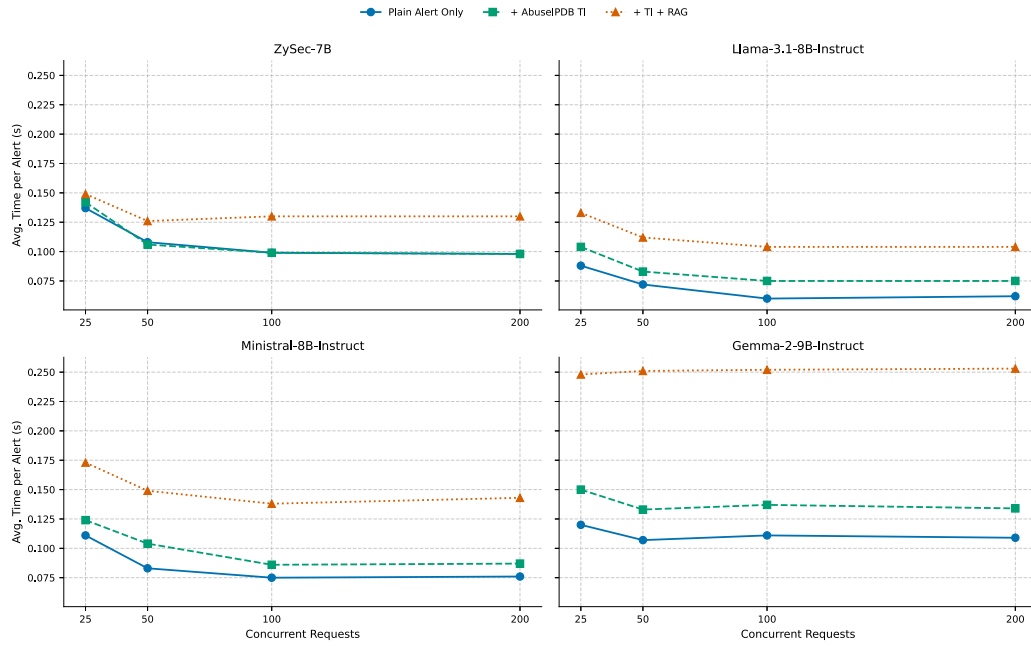


Fig. 7. LLMs inference times (avg. per alert) for different number of concurrent requests.

Therefore, the FPR reduction should be contextualized as a valuable, but secondary, benefit, with the primary worth being the collaborative knowledge transfer for unseen attacks.

5.4. Inference time analysis

This experiment was conducted to analyze the different LLMs inference times, which is a key performance metric given the real-time nature of the proposed solution. Since the models allow processing several user inputs in parallel, different number of concurrent alerts were tested to assess the performance of each model under varying load conditions, simulating a realistic environment where alerts would be ingested at different rates, i.e., high rates when an attack is taking place in the monitored network, and low rates when only benign traffic is being generated. To this end, the data from the BFA re-evaluation experiment was used, consisting of 342 low-confidence alerts to be ingested to the LLMs.

Fig. 7 illustrates the impact of increasing concurrency on the average time per alert for each LLM and context setting. A key observation across all models is the benefit of batching; as the number of concurrent requests increased from 25 to 100, the average inference time per alert generally decreased. This demonstrates that the vLLM inference engine effectively utilizes the GPU to achieve higher throughput under heavier loads. Beyond 100 concurrent requests, the performance tends to maintain or even degrade, indicating that the system has reached a point of optimal throughput. Based on this finding, the 100-request mark was chosen as a representative point for a detailed performance comparison.

In Fig. 8, a direct comparison of the models fixing the number of concurrent requests to 100 is presented. From the results, two main conclusions can be extracted. First, the depth of context directly impacts inference time. For every model, adding external TI from AbuseIPDB increased the processing time, and the further addition of inter-organizational TI from the Federated RAG Incident Store resulted in the longest inference times. This was expected, as a richer context results in a longer prompt for the LLM to process. The most significant time penalty was associated with the full context setting, with *Gemma-2-9B-Instruct* showing the most dramatic increase, rising from 0.137s to 0.252s. Second, there were clear performance differences among the evaluated models. *Llama-3.1-8B-Instruct* was the fastest overall, processing a plain alert in just 0.060s. *Ministral-8B-Instruct* also demonstrated excellent performance, taking only 0.075s for a plain alert and 0.138s

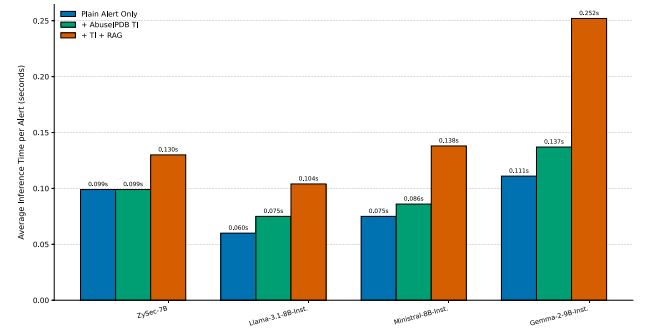


Fig. 8. LLMs average inference times (avg. per alert) for each context configuration, with fixed 100 concurrent requests.

for the full context. An interesting anomaly was observed with *Zysec-7B*, where adding the AbuseIPDB TI context resulted in no measurable latency increase (0.099s for both settings). This behavior is likely linked to the model's previously noted reliability issues; the time cost of the longer prompt was probably offset by the format-enforcing backend prematurely terminating generations whenever the model failed to produce a valid output that adhered to the required output format. Despite these variations, the results are highly encouraging. Even with the full context, the best-performing models maintained inference times in the order of 100–140 milliseconds per alert. This demonstrates that the proposed framework is not only effective at improving detection but is also sufficiently fast for deployment in real-time or near-real-time security monitoring applications.

5.5. Ablation study: Comparison with a traditional classifier

The results of the comparative ablation study are presented in Fig. 9, which compares the recall of the four LLMs against the traditional XGBoost and Random Forest models on the BFA re-evaluation task, all using the full context setting. The analysis shows that the traditional models performed exceptionally well, with XGBoost achieving a high recall of 98.73%. However, the top-performing LLMs (*Gemma-2-9B-Instruct* and *Ministral-8B-Instruct*) demonstrated a quantitative advantage by achieving a perfect 100% recall.

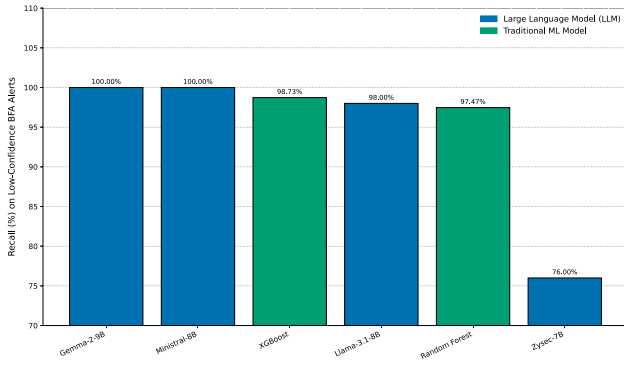


Fig. 9. LLMs vs. traditional ML models recall in classifying BFA low-confidence alerts when given full context.

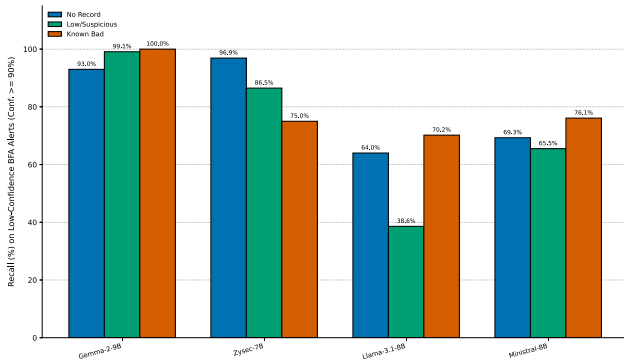


Fig. 10. LLMs recall in classifying BFA low-confidence alerts with varied external TI quality.

While this slight increase is valuable in a security context, the primary justification for employing an LLM extends beyond pure classification accuracy. Unlike traditional models, an LLM provides crucial narrative explainability, generating human-readable rationales for its decisions that are essential for analyst trust and rapid response. It also offers superior flexibility, adapting to new, unstructured data sources via prompt changes (e.g., additional context injection) rather than requiring a full model retraining cycle and a process of maintaining and updating a high-quality dataset. Finally, as a reasoning engine, it can perform complex task chaining, such as suggesting mitigation steps based on the identified threat, providing additional value to the attack management loop.

Therefore, while a traditional model is an efficient classifier to be used for the re-evaluation task, the LLM's demonstrable performance edge and its unique advantages in explainability and reasoning justify its higher computational cost. These capabilities are what elevate the framework beyond a simple FL-based intrusion detection system into a true federated intelligence framework that provides richer, more actionable security insights.

5.6. Robustness evaluation under ambiguous external threat intelligence

The results of the experiment evaluating LLM performance with varied external TI quality are presented in Fig. 10. The figure shows each model's recall on the BFA re-evaluation task. Specifically, the percentage of BFA alerts originally misclassified by the base model that were correctly re-classified by the LLM with a confidence of 90 % or higher.

In the "No Record" scenario, *Gemma-2-9B-Instruct* and *Zysec-7B* achieved excellent recall (93.0 % and 96.9 %, respectively) by relying primarily on the shared knowledge from the RAG store. This shows the

system can produce reliable, high-confidence corrections even without external signals.

In the "Low/Suspicious" scenario, *Gemma-2-9B-Instruct* maintained exceptionally high confidence (99.1 % recall), demonstrating its ability to prioritize the strong RAG context over weak or contradictory external signals. Conversely, the rest of the evaluated models showed a performance drop, with *Llama-3.1-8B-Instruct* experiencing the biggest one as recall fell to 38.6 %, indicating greater sensitivity to conflicting evidence.

This experiment revealed significant differences in model robustness compared to the initial evaluation (see Section 5.2, Fig. 3). *Gemma-2-9B-Instruct* consistently demonstrated superior robustness and confidence across both tests. In contrast, *Ministral-8B-Instruct*, which achieved perfect recall in the initial test with strong TI, showed a marked decrease in performance here (65-76 % recall), suggesting its effectiveness may be more dependent on clear, unambiguous and consistent context, since in prior experiments a single, static attacker external TI profile was employed. Perhaps most notably, *Zysec-7B*, the worst performer in the initial test, excelled in this experiment, particularly in the "No Record" scenario. This suggests its domain-specific tuning makes it highly adept at reasoning from core technical data, i.e., traffic flow features. However, this further illustrates the fine-tuning trade-off discussed in Section 5.3. Its comparatively lower performance in the "Known Bad" scenario (75 % recall) here, combined with its poor performance when given full context in the initial test, indicates it may be less effective than general-purpose models at integrating and weighing multiple, diverse information sources simultaneously. When strong external TI is added to the flow and RAG context, its specialized analysis might be hindered or lead to lower confidence compared to models with broader reasoning capabilities which, combined with its format reliability issues, makes it a questionable choice for this context-fusion task. Finally, regarding *Llama-3.1-8B-Instruct*, the information provided in the "Known Bad" scenario did not suffice to pair its performance in the initial experiment, showing weakness against varying external TI similarly to *Ministral-8B-Instruct*.

Overall, this nuanced test provides strong evidence that the two-tiered TI system, particularly the Federated RAG component, enables robust performance even when external intelligence signals are imperfect, validating the framework's design for more realistic conditions, and specifically assessing the RAG component's value in scenarios where external TI is unavailable or ambiguous. Furthermore, it highlights the importance of evaluating LLMs under varied conditions, as performance on idealized benchmarks may not fully predict robustness in real-world scenarios with ambiguous data.

6. Challenges & limitations

During the design and evaluation of the solution, several significant challenges and limitations were identified. This section describes them, discusses their potential impact, and proposes countermeasures for mitigation.

6.1. Threat model and security analysis

The framework's threat model distinguishes between unintentional risks (e.g., misclassifications arising from normal operation) and intentional, adversarial attacks (e.g., a malicious participant attempting to corrupt the system). The following subsections analyze these distinct threats.

6.1.1. Model poisoning via incorrect feedback

The system's retraining loop, which uses LLM-verified alerts as new training data to improve the base model via subsequent FL training, introduces a potential vector for model poisoning. The risk is that a sample incorrectly classified by either the base detection model or the LLM with high confidence would be added to the training set and thereby used

to train future iterations of the detection model. For instance, if traffic flows associated with a sophisticated low-profile attack are misclassified as benign, the model would fail to learn the new threat. More critically, if benign flows are misclassified as any attack type with high confidence, these poisoned samples would be used blindly for retraining. Over time, the injection of such data could degrade the model's performance.

To mitigate this critical threat, several countermeasures can be implemented. The first relies on raising the confidence thresholds, reducing the risk of including erroneous samples at the cost of having less data for retraining. Another potential technique is to include a human-in-the-loop for verification, where alerts are reviewed by a security analyst before being permanently added to the Local Incident Store, which would reduce misclassifications at the cost of added latency. Lastly, a periodic audit mechanism could be integrated to evaluate the base model's performance over time on a static, trusted dataset. If a performance drop is detected, the most recently added training samples could be quarantined for subsequent review, and the model could be rolled back to a previous, trusted version to ensure the framework remains fully operational while this analysis takes place.

6.1.2. Federated RAG incident store poisoning

The inter-organizational threat intelligence information exchange based on a Federated RAG Incident Store relies on the honesty of all participants. Potentially, a malicious or compromised organization within the federation could intentionally upload misleading incident summaries. For example, they could share benign incidents as novel attacks or mislabel a known attack type. This would poison the shared knowledge base, potentially causing other organizations' LLMs to make incorrect classifications based on this faulty intelligence information being added as context.

To address this threat, two main countermeasures could be implemented. First, a federated reputation and trust mechanism could be established, where each participating organization has a trust score which is continuously updated based on its historical behavior and contributions. The organization's trust score would be added to each incident shared, so that the different organizations' LLMs could weigh the context from that source accordingly, potentially reducing its reliance on information from low-trust participants. If an organization's trust score drops below a pre-established threshold, it could be removed from the federation. However, this mechanism practical implementation faces significant unsolved challenges, including: (i) rigorously quantifying historical behavior into a non-gameable metric, (ii) defending against Sybil attacks, where a single adversary creates multiple fake identities, (iii) preventing collusion attacks, where multiple participants conspire to build trust before submitting poisoned data, and (iv) solving the "cold-start" problem for new organizations, whose valuable contributions might be initially down-weighted.

Second, an anomaly detection mechanism could be implemented within the inter-organization domain, to detect incident embeddings drastically different from all previously seen embeddings for that attack class. In that case, a review process would take place before adding the incident to the shared incident store. Furthermore, this system would introduce a significant bottleneck, requiring dedicated human resources to manually review the flagged incidents.

6.1.3. Adversarial FL model poisoning

In addition to the previous challenges, a more direct threat involves an adversarial participant intentionally submitting poisoned model updates during the FL training process itself to corrupt the global model. While achieving Byzantine fault tolerance to defend against such threats is a general research challenge for all FL systems, a framework for cybersecurity must acknowledge it. The proposed framework, which uses standard aggregation algorithms (e.g., FedAvg, FedAdam), does not currently implement specific defenses (e.g., robust aggregation algorithms, anomaly detection on client updates) against such attacks. This remains a critical limitation and a priority for future work.

6.2. LLM Operational challenges

6.2.1. Local LLM as a bottleneck under heavy load

As demonstrated in Section 5.4, fastest LLMs among the evaluated set achieve inference times per alert close to 100 milliseconds. Although the achieved performance could be considered near optimal or state-of-the-art given the deployment setup and the hardware used (see Section 4.3), the inference times are still orders of magnitude behind the performance of the base detection model, due to the vast difference between the two model types. During a large-scale event, such as a DDoS attack or a network-wide scan, where a massive amount of traffic is processed, the Real-time Detection Engine would produce alerts at a rate the local LLM can not match. Even though the first confidence filtering is implemented partly for mitigating this, and would discard a significant portion of the samples for re-evaluation, the LLM would still represent a processing bottleneck, preventing the near-real-time re-evaluation of potentially critical alerts.

Two main countermeasures are proposed to mitigate this limitation of the framework. The first is based on dynamically adjusting the first confidence threshold for sending alerts to the LLM. During low traffic volume periods, the threshold could be kept lower to analyze more alerts and potentially create more samples for retraining, while during a high-volume period, the framework load could be monitored and the threshold could be raised automatically to ensure only the most ambiguous alerts are sent for re-evaluation, preserving the LLM's responsiveness. In addition, a second countermeasure could implement priority queues. Alerts generated by the Real-time Detection Engine would be prioritized based on factors such as the potential severity of the initially predicted attack or the criticality of the targeted asset, ensuring that the most important alerts are included in the highest priority queue and thereby analyzed first. Validating these proposed mitigation strategies under a high-volume, large-scale stress test designed to quantify queueing delays and potential alert drops is an important and necessary direction for future work.

6.2.2. LLM Hallucination and unreliability

A fundamental challenge for any system leveraging LLMs in security-critical applications is the inherent risk of model hallucination. In this regard, our framework was designed with two primary architectural defenses to minimize this risk.

The first defense is the use of RAG itself. The LLM's reasoning is explicitly grounded in the factual and real-time context provided in the prompt (traffic flow features, TI report, and RAG results), rather than its parametric memory, substantially reducing the likelihood of generating unsupported information. The second mitigation mechanism is structured output enforcement via a guided decoding backend. This constrains the LLM's output to a strict format, preventing unstructured responses and ensuring its output is in most cases machine-parsable and directly addresses the classification task. However, our experiments concluded that, even enforcing this countermeasure, some models (specifically, the Zysec-7B) still produced outputs which did not align with the required format.

Despite these significant mitigations, we acknowledge that the risk of subtle factual inaccuracies or flawed reasoning is not entirely eliminated. To further harden the framework against this residual risk in future work, several advanced techniques could be explored. These include: (i) automated fact-checking, implementing a secondary verification layer that cross-references the LLM's key assertions against other independent data sources before an alert is finalized, (ii) querying a diverse set of LLMs for the same task and flagging alerts for human review if there are significant disagreements in their conclusions, and (iii) confidence calibration through applying techniques to the LLM's raw confidence scores to ensure they more accurately represent the true probability of the prediction being correct, making the confidence filter more reliable.

Ultimately, given the current state of LLM technology, our framework is designed to act as a powerful analysis-assist tool. For critical incidents, human oversight remains the final validation step.

6.2.3. Handling conflicting threat intelligence

Beyond hallucinations, a key operational challenge is handling conflicting information from multiple TI sources (e.g., one service reports "malicious" while another reports "benign"), an aspect our study did not evaluate. Developing a robust fusion engine to weigh and resolve such conflicts, or enabling the LLM to reason about the disagreement, is a significant area for future research.

6.3. Scope of experimental validation

6.3.1. Robustness to novel zero-Day threats

While the framework is designed for participating organizations to learn unseen attacks via the feedback loop, the detection of a truly novel threat, i.e., a zero-day attack, remains a limitation. It is important to distinguish between two types of novel threats: those from new infrastructure using known techniques, and those employing a truly novel technique (i.e., a zero-day exploit).

The framework demonstrates significant resilience against the first type. Our previous robustness analysis (see Table 10) simulated this in the "No Record" scenario, where external TI sources had no record of the attacker. In that situation, the framework successfully identified the attack by leveraging the shared knowledge of similar attack patterns in the Federated RAG Incident Store. This proves the system's value in handling emerging threats that reuse known TTPs from new sources. Furthermore, as observed with Zysec-7B in prior experiments (see Section 5.3), some models can interpret this general lack of information as suspicious, though this is not a guaranteed or reliable detection method on its own.

However, for a truly novel, zero-day technique for which no similar incidents exist in the RAG store, the TI context would be completely limited, and the system would likely fail. The resulting traffic would be misclassified by the base detection model. Even if it is predicted with a confidence below the first threshold, which would trigger the re-evaluation pipeline based on the local LLM, the absence of reported malicious behaviour in either of the two tiers of TI would make achieving a precise classification difficult.

To remediate this, periodic unsupervised anomaly detection could be used in the framework to continuously learn benign behavior, i.e., from benign traffic flows, and alert when a statistically rare flow is detected, regardless of any specific attack type. These anomalies could be either sent to the LLM or evaluated by a security analyst to discover zero-day attacks. However, this would require training and maintaining a parallel model that, although it could be also trained through FL with other organizations and consume the same information as the supervised mechanism, i.e., traffic flow statistics, it introduces significant complexity.

In addition, incorporating further TI sources that specifically focus on newly discovered vulnerabilities and zero-day exploits could provide the LLM with the necessary context to identify and correctly classify zero-day attacks. This would align with the previous countermeasure, as the LLM could effectively evaluate the alerts generated by the anomaly detection model. While this framework's primary focus is on detection and analysis, it contributes to prevention by providing human-readable explanations that enable security analysts to validate threats and deploy mitigations more rapidly. Nonetheless, the potential subscription costs of such TI feeds could pose a challenge when implementing this mitigation.

6.3.2. Dataset timeliness and threat representation

Our study deliberately employed the CIC-IDS-2017 dataset [26], a foundational benchmark in intrusion detection research. This was a conscious methodological trade-off. While acknowledging the dataset's

age, its high-quality, labeled data and diverse set of well-defined attack classes were essential for constructing the clear, controlled, and measurable non-IID scenario needed to validate our novel architectural framework. Furthermore, using this widely-recognized benchmark ensures our baseline results are comparable within the existing literature. We prioritized this structural clarity and labeling quality over more recent datasets which, while capturing newer threats, often lack the granularity or consistency required for this specific type of architectural validation.

Consequently, the scope of our current validation is focused on the types of network attacks represented within this benchmark. We acknowledge that, as a result of this deliberate focus, the framework's performance against more contemporary threats-such as supply chain attacks, container-based exploits, or threats within modern encrypted traffic-was not evaluated in this study. Therefore, extending this validation to more recent datasets that incorporate these evolving threats is an important and logical next step, which can build upon the architectural principles established and validated in this work.

6.3.3. Scalability to larger federations

The use of only two organizations in our experiment represents a simplified scenario. While this was necessary for a controlled validation of our core concept, it does not address the challenges of deploying the framework in a larger federation of dozens or hundreds of participants. Real-world scalability would introduce additional complexities, such as increased communication overhead for the FL aggregator, managing trust and data quality from a larger and more diverse group of organizations, or handling potential network latency issues affecting real-time TI queries. This also includes communication efficiency challenges, as our study did not evaluate model compression, gradient quantization, or other optimizations necessary for deployment in resource-constrained environments with high latency or low bandwidth. Furthermore, the framework's reliability under common operational challenges, such as network partitions or intermittent connectivity, was not evaluated, and remains a key area for future investigation.

6.4. Practical deployment considerations

While the proposed framework is effective, its implementation presents significant challenges in terms of cost and complexity. Although the exact high-end hardware used in our evaluation may not be required for all real-world deployments, as it was chosen to establish a strong performance benchmark for the framework's inference capabilities, rather than to suggest a minimum deployment requirement, powerful computational resources, particularly GPUs, are still strongly recommended to achieve near-real-time performance. This hardware requirement, combined with the sophisticated software stack and the technical expertise needed to deploy and maintain these systems, creates a high barrier to entry. Consequently, the solution in its current form may be impractical for smaller organizations with limited budgets or personnel.

To address this, two potential future directions could be explored. One approach is to offer the LLM re-evaluation pipeline as a centralized cloud service that organizations could query via an API. This would abstract away the hardware and maintenance complexity for participants. However, this model introduces its own critical trade-offs. While initial capital costs would be lower, long-term operational costs could be higher. More importantly, it would alter the framework's decentralized nature and introduce significant data privacy and trust considerations, as organizations would need to send potentially sensitive data to a third-party provider.

Another, more decentralized solution, is to focus on model distillation and quantization. This would involve creating smaller, highly optimized versions of the powerful LLMs that can run efficiently on lower-end, commodity hardware. This approach would significantly lower the deployment cost for all participants while preserving the self-hosted architecture, but it would likely come at the expense of some degradation

in accuracy. However, this entire challenge is expected to lessen over time, given the rapid industry trend toward producing smaller, yet increasingly capable and efficient, models with each new generation.

7. Conclusion

This paper addressed the significant performance degradation of FL for IDS in realistic non-IID environments. We proposed a novel framework where a local LLM, whose context is augmented by both external and collaborative, privacy-preserving threat intelligence via a Federated RAG architecture, enriches and re-evaluates the low-confidence alerts produced by the base, FL-trained detection model. The experimental results demonstrated the framework's powerful capabilities. The feedback loop using LLM-verified data dramatically improved the base model's performance on a missing attack class, boosting recall from 75.85 % to 91.47 % and the F1-score from 86.27 % to 95.54 %. Furthermore, the solution proved practical, reducing the overall FPR while maintaining inference times (~100-250ms on our testbed) that, while orders of magnitude slower than the base model, are suitable for the near-real-time re-evaluation of critical, low-confidence alerts. This work demonstrates a conceptual shift from simple federated learning to a more comprehensive "federated intelligence" paradigm, where organizations can share rich, contextual knowledge securely and effectively. While acknowledging practical challenges such as potential model poisoning and deployment complexity, future work will focus on developing federated trust mechanisms and creating more efficient, distilled models, thereby making the framework accessible to smaller, budget-constrained organizations. Ultimately, this research represents a significant step towards a new generation of more effective, collaborative, and explainable cybersecurity defenses.

CRedit authorship contribution statement

Pablo Fernández Saura: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing - original draft, Writing - review & editing; **Jorge Bernal Bernabé:** Conceptualization, Investigation, Writing - original draft, Writing - review & editing; **Antonio Skármeta Gómez:** Writing - original draft, Supervision, Resources, Project administration, Funding acquisition.

Data availability

Data will be made available on request.

Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was funded by the Horizon projects RIGOUROUS (GA: 101095933) and ResilMesh (GA: 101119681), and by the Spanish INCIBE's 6G-SOC project through the European Union's NextGenerationEU Recovery, Transformation, and Resilience Plan.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, B. Aguera y Arcas, Communication-Efficient Learning of Deep Networks from Decentralized Data, in: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, 2017, pp. 1273–1282. <https://doi.org/10.48550/arXiv.1602.05629>

[2] S. Pandey, H. Azath, R.U. Rahman, H. Lamkuche, Privacy-Preserving model for cyber threat intelligence, sharing across multi-Organizational platforms, in: 2025 IEEE 14th International Conference on Communication Systems and Network Technologies (CSNT), 2025, pp. 437–442. <https://doi.org/10.1109/CSNT64827.2025.10968450>

[3] J.G. Daniel M., D. Solans, M. Heikkilä, A. Vitaletti, N. Kourtellis, A. Anagnostopoulos, I. Chatzigiannakis, Non-IID data in Federated Learning: A Survey with Taxonomy, Metrics, Methods, Frameworks and Future Directions, preprint on arXiv (2024). <https://doi.org/10.48550/arXiv.2411.12377>

[4] X. Ma, J. Zhu, Z. Lin, S. Chen, Y. Qin, A state-of-the-art survey on solving non-IID data in federated learning, Future Gener. Comput. Syst. 135 (2022) 244–258. <https://doi.org/10.1016/j.future.2022.05.003>

[5] T. Li, A.K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, V. Smith, Federated Optimization in Heterogeneous Networks, in: Proceedings of Machine Learning and Systems, 2020, pp. 429–450. <https://doi.org/10.48550/arXiv.1812.06127>

[6] S.P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, A.T. Suresh, SCAFFOLD: Stochastic controlled averaging for federated learning, in: Proceedings of the 37th International Conference on Machine Learning, 119 of Proceedings of Machine Learning Research, 2020, pp. 5132–5143. <http://proceedings.mlr.press/v119/karimireddy20a/karimireddy20a.pdf>

[7] S. Reddi, Z.B. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, B. McMahan, Adaptive Federated Optimization, in: International Conference on Learning Representations, 2021. <https://doi.org/10.48550/arXiv.2003.00295>

[8] Y. Liu, G. Wu, W. Zhang, J. Li, Federated learning-Based intrusion detection on non-IID data, in: Algorithms and Architectures for Parallel Processing, 2023, pp. 313–329. https://doi.org/10.1007/978-3-031-22677-9_17

[9] D. Hamouda, M.A. Ferrag, N. Benhamida, H. Seridi, Advancing cybersecurity with LLMs: a comprehensive review of intrusion detection systems and emerging applications, in: Proceedings of the 7th International Conference on Informatics and Applied Mathematics (IAM 2024), 2025, pp. 70–80.

[10] Y. Chen, M. Cui, D. Wang, Y. Cao, P. Yang, B. Jiang, Z. Lu, B. Liu, A survey of large language models for cyber threat detection, in: Comput. Secur., 2024, pp. 104016145. <https://doi.org/10.1016/j.cose.2024.104016>

[11] J. Zhang, H. Bu, H. Wen, Y. Liu, H. Fei, R. Xi, L. Li, Y. yang, H. Zhu, D. Meng, When LLMs meet cybersecurity: a systematic literature review, in: Cybersecur, 2025, pp. 558. <https://doi.org/10.1186/s42400-025-00361-w>

[12] L. Galeano, J. Jimenez, J. Schafer, I. MedinaBulo, LLM-Based Cyberattack detection using network flow statistics, in: Appl. Sci., 2025, pp. 652915. <https://doi.org/10.3390/app15126529>

[13] E. Nwafor, U. Baskota, M.S. Parwez, J. Blackstone, H. Olufowobi, Evaluating large language models for enhanced intrusion detection in internet of things networks, in: GLOBECOM 2024 - 2024 IEEE Global Communications Conference, 2024, pp. 3358–3363. <https://doi.org/10.1109/GLOBECOM52923.2024.10901300>

[14] Y. Otoum, A. Asad, A. Nayak, LLM-Based Threat Detection and Prevention Framework for IoT Ecosystems, preprint on arXiv (2025). <https://doi.org/10.48550/arXiv.2505.00240>

[15] M.M. Rahman, M.S. Ullah, S. Nahar, M. Rahman, M.M. Rahman, The role of explainable AI in cyber threat intelligence: enhancing transparency and trust in security systems, World J. Adv. Res. Rev. 23 (2024) 2897–2907. <https://doi.org/10.30574/wjarr.2024.23.2.2404>

[16] B. Desai, K. Patil, I. Mehta, A. Patil, Explainable AI in cybersecurity: a comprehensive framework for enhancing transparency, trust, and human-AI collaboration, in: 2024 International Seminar on Application for Technology of Information and Communication (iSemantic), 2024, pp. 135–150. <https://doi.org/10.1109/iSemantic63362.2024.10762690>

[17] A. Bilal, D. Ebert, B. Lin, LLMs for Explainable AI: A Comprehensive Survey, preprint on arXiv (2025). <https://doi.org/10.48550/arXiv.2504.00125>

[18] K. Kurniawan, A. Kiesling, CyKG-RAG: towards knowledge-graph enhanced retrieval augmented generation for cybersecurity, in: RAGE-KG 2024 Workshop at ISWC 2024, 2024, pp. 51–64. <https://ceur-ws.org/Vol-3950/paper1.pdf>

[19] F. Loumachi, M. Ghanem, M. Ferrag, Advancing cyber incident timeline analysis through retrieval-Augmented generation and large language models, Comput (2025). 67214. <https://doi.org/10.3390/computers14020067>

[20] A. Lekssays, U. Shukla, H.T. Sencar, M.R. Parvez, TechniqueRAG: retrieval augmented generation for adversarial technique annotation in cyber threat intelligence text, in press in: Findings of the Association for Computational Linguistics (ACL 2025), 2025.

[21] P.F. Saura, K.R. Jayaram, V. Isahagian, J.B. Bernabé, A. Skarmeta, On automating security policies with contemporary LLMs, in: 2025 IEEE International Conference on Software Services Engineering (SSE), 2025, pp. 77–83. <https://doi.org/10.1109/SSE67621.2025.00018>

[22] Ismail, R. Kurnia, F. Widyatama, I.M. Wibawa, Z.A. Brata, Ukasyah, G.A. Nelistiani, H. Kim, Enhancing security operations center: wazuh security event response with retrieval-Augmented-Generation-Driven copilot, Sensors (2025). 87025 <https://doi.org/10.3390/s25030870>

[23] A. Chakraborty, C. Dahal, V. Gupta, Federated Retrieval-Augmented Generation: A Systematic Mapping Study, preprint on arXiv (2025). <https://doi.org/10.48550/arXiv.2505.18906>

[24] L. Ammann, S. Ott, C.R. Landolt, M.P. Lehmann, Securing RAG: A Risk Assessment and Mitigation Framework, preprint on arXiv (2025). <https://doi.org/10.48550/arXiv.2505.08728>

[25] B. Zhang, H. Xin, M. Fang, Z. Liu, B. Yi, T. Li, Z. Liu, Traceback of poisoning attacks to retrieval-Augmented generation, in: Proceedings of the ACM on Web Conference 2025, 2025, p. 2085-2097. <https://doi.org/10.1145/3696410.3714756>

- [26] I. Sharafaldin, A.H. Lashkari, A.A. Ghorbani, Toward generating a new intrusion detection dataset and intrusion traffic characterization, in: International Conference on Information Systems Security and Privacy, 2018, pp. 108–116. <https://doi.org/10.5220/0006639801080116>
- [27] ZySec-AI, SecurityLLM (Revision 7980811) [software], Hugging Face, 2025. <https://huggingface.co/ZySec-AI/SecurityLLM>.
- [28] Meta-AI, Llama-3.1-8B-Instruct [software], Hugging Face, 2024. <https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct>.
- [29] Mistral-AI, Ministral-8B-Instruct-2410 [software], Hugging Face, 2024. <https://huggingface.co/mistralai/Ministral-8B-Instruct-2410>.
- [30] Google, Gemma-2-9b-it [software], Hugging Face, 2024. <https://huggingface.co/google/gemma-2-9b-it>.



Pablo Fernández Saura is a PhD candidate in Computer Science at the University of Murcia, Spain, where he also received his B.S. and M.S. degrees. As a researcher at the university, he has contributed to several national and European cybersecurity projects, including RIGOUROUS, RESILMESH, CyberSec4Europe, and INSPIRE-5G. His international experience includes a role as a Visiting Researcher with IBM at the Thomas J. Watson Research Center. His research interests focus on AI-assisted intrusion detection, next-generation networks, federated learning, and LLMs, fields in which he has authored several publications for high-impact journals and conferences.



Jorge Bernal Bernabé received the B.S., M.S., and Ph.D. degrees in computer science and the M.B.A. degree from the University of Murcia, Spain, where he is currently an Associate Professor. He has been a Visiting Researcher with the Hewlett-Packard Laboratories and University of the West of Scotland. He has authored over 40 high-impact indexed journal papers and numerous conferences. During the last years, he has been working in several European research projects, such as SocIoTal, ARIES, OLYMPUS, ANASTACIA, INSPIRE-5G and CyberSec4EU. His scientific activity is mainly devoted to the security, trust, and privacy management in distributed systems and IoT.



Antonio F. Skarmeta is a full professor at the University of Murcia, the Department of Information and Communications Engineering. His research interests are the integration of security services, identity, the Internet of Things, and smart cities. Skarmeta received a Ph.D. in computer science from the University of Murcia. He has published more than 200 international papers and been a member of several program committees.