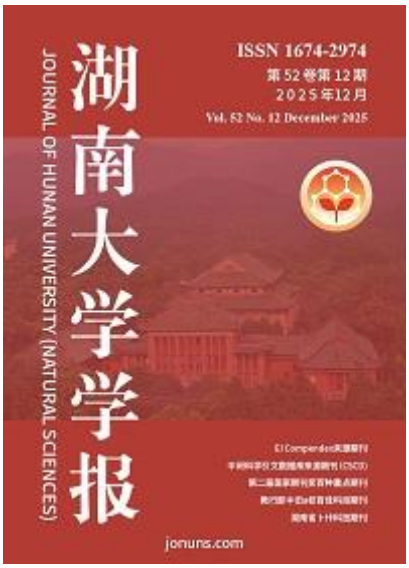

Journal of Hunan University

(Natural Sciences)

Vol. 52 No. 12

December 2025

Available online at

https://jonuns.com

Open Access Article

https://doi.org/10.55463/issn.1674-2974.52.12.1

Generative AI–Guided Sentinel for Self-Optimizing Federated Cybersecurity and Intelligent Threat Detection

Akter Rokaya[1], Md Al Samiul Amin Rishat[2], Singh Sudhanshu[1], Abhishank Singh[1], Bian Naizheng[1,*]

[1] Computer Science and Electrical Engineering, Hunan University, Changsha, China;

[2] Department of Information Science, University of North Texas (UNT), Denton, USA;

* Corresponding author: nbian@hnu.edu.cn

Article history
Received: November 15, 2025
Revised: December 26, 2025
Accepted: January 5, 2026
Published: January 30, 2026

Abstract: As cyber threats become increasingly sophisticated and pervasive, adaptive, intelligent, and privacy-preserving intrusion detection systems (IDSs) are more critical than ever, particularly in ecosystem-based networks. However, existing federated cybersecurity systems continue to face several challenges, including reduced data processing efficiency caused by data heterogeneity, difficulties in real-time threat detection, and complex configuration management.

To address these challenges, we propose a novel Self-Improving Federated Cybersecurity Sentinel framework that integrates Federated Learning (FL) with Generative Artificial Intelligence (GAI) to enable dynamic and context-aware optimization. Large Language Models (LLMs) play a central role in the proposed framework by enabling prompt-driven analytics for real-time intrusion detection, feature relevance assessment, automated anomaly investigation, and adaptive optimization of FL parameters.

Experimental evaluations conducted on the UNSW-NB15 dataset demonstrate that the proposed framework achieves a precision of 0.96 and an F1-score of 0.98, while simultaneously reducing configuration adjustment time by approximately 68%. These results indicate that the framework significantly simplifies the tuning process and enhances detection performance. Overall, this study represents a substantial advancement toward fully autonomous

and robust cybersecurity systems. Future work will focus on improving the generalizability of the framework across diverse threat scenarios and incorporating explainable AI components to enhance transparency and interpretability.

Keywords: Federated Learning, Large Language Model, Cybersecurity, Intrusion Detection, Generative AI.

生成式人工智能引导的哨兵系统，用于自我优化的联邦网络安全与智能威胁检测

摘要：随着网络威胁不断演化，在分布式环境中构建自适应、智能化且具备隐私保护能力的入侵检测系统愈加关键。然而，现有的联邦式网络安全框架仍受到数据异质性、实时检测能力不足以及配置管理复杂等因素的限制。为解决这些问题，本研究提出“自进化联邦式网络安全哨兵”框架，将联邦学习（FL）与生成式人工智能（GAI）相结合，实现动态、上下文感知的优化机制。在该框架中，大语言模型（LLM）发挥核心作用，可通过提示驱动的分析支持实时入侵检测、特征重要性评估、自动化异常判断以及联邦学习参数优化，从而增强系统对复杂威胁的响应能力。基于 UNSW-NB15 数据集的实验表明，该框架达到 0.96 的精确率与 0.98 的 F1-Score，并将配置调整时间减少约 68%，显示出在性能与效率上的明显优势。研究结果说明该框架能有效提升威胁检测质量并简化调优流程。未来工作将致力于增强框架在多类威胁场景下的泛化能力，并引入可解释人工智能模块，以提高系统透明度与可信度。

关键词：联邦学习、大语言模型、网络安全、入侵检测、生成式人工智能

1. Introduction

Network intrusion is a growing concern in modern cybersecurity. It is defined as “an unauthorized approach to the data within a network system to compromise the confidentiality of the system” [1]. Intrusion detection systems (IDS) actively look for a wide range of malicious behaviors, such as malware, DoS attacks and attempts to gain unauthorized access. This helps lessen the effects of cyber threats [2]. Most network-based IDS use sensors to look at network packets, analyzers to find strange patterns, and user interfaces to give cybersecurity teams complete logs and alerts as soon as possible [3]. As a result, IDS technologies have developed to include advanced anomaly detection methods, often enhanced by machine learning, to effectively identify new cyber threats [4].

“FL is a distributed machine learning technique that improves privacy and security by allowing multiple nodes to work together to train a shared model without sharing raw data” [5]. The idea of aggregating model updates instead of directly sending raw data alleviates the privacy issues and minimizes the risk of centralized data exposure, which makes FL particularly promising for privacy-sensitive applications like critical infrastructure protection, health care and finance [6]. The expanding application of AI-enabled tools, such as high-accuracy classroom chatbots reveals the potential for intelligent automated systems to transform numerous sectors, not least cybersecurity [7]. Recent studies have demonstrated the effectiveness of FL in significantly advancing network intrusion detection using diverse, geographically distributed datasets under privacy constraints [8].

There are advantages to FL-based intrusion detection methods. However, these techniques currently encounter significant challenges. First, FL models are hard to understand and explain, which makes it hard to figure out what went wrong and give cybersecurity analysts useful information [9]. Second, optimizing is even harder in federated situations because it requires organizing data that are different from each other [10]. Thirdly, a lot of current FL oriented processes require people to look at the output data by hand and/or set hyperparameters by hand before they can be tested with settings other than “out of the box” [11]. This is likely to make the cybersecurity analyst’s job harder when it comes to analysis and make it harder to make it available for use by more people. So, to make FL-based intrusion detection models better, we need new ways to automate, understand, and offer a stronger solution.

To advance the practicality and robustness of federated learning, this study seeks to incorporate sophisticated Large Language Model (LLM) analytics into a federated learning (FL) framework for network intrusion detection. Ultimately, this kind of synergy will lead to better FL-based cybersecurity systems, making them more resilient overall, more efficient in operation and easier to understand and explain. By utilising LLMs, our proposed system yields model-generated insights, hyperparameter optimization, and human-readable or syntactically descriptions, connecting complex machine

learning generation to actionable decisions in a distributed security domain. The mentioned items are the main contributions:

- LLM Driven Analytics and Optimization: We developed a module driven by LLM that automatically checks the network traffic patterns, key features, potential anomalies, hyperparameters for federated learning and security alert changes. The module is designed to execute strictly on the server, and won't consume any additional computing power from clients.
- Prompting Engineering and Checking Output: For llm, we had a complete prompt engineering document that gives the acceptable output format with examples and instructions. This is simply to ensure that the LLM returns structured, parseable JSON outputs. The LLM responses will be returned and we will test that the LLM responders are being constructed properly. Even if API processing produces poor outputs, the fallback logic will ensure a strong system.
- Automated Explainability and Actionable Insights: The LLM module will produce interpretable, actionable explanations and recommendations for people that bridge raw model outputs to operational actions. This will be beneficial in making the FL system more visible to both technical and non-technical stakeholders.
- The Analysis of the LLM System: We will check how the system can be analyzable in the LLM module and how robust it is. This will demonstrate how practical the module is in real privacy-preserving FL deployment scenarios.

2. Literature Review

Intrusion Detection Systems (IDS) are a key part of modern cybersecurity systems. There have been standard setting tools that use rule and behavior based analysis of network traffic, such as Snort [12], Bro/Zeek, and Suricata [13]. However, this process has moved on to more advanced detection pipelines, as shown by Landauer et al. [14] use sensor arrays, feature extraction modules, and an alert generation framework to keep an eye on, find, and respond to strange activities. Current methodologies employing datasets such as UNSW-NB15, KDD99, and NSL-KDD have trained learners utilizing ensemble techniques, decision trees, and support vector machines (SVMs) [15]. Later methods, like DeepIDS [16], aimed to teach deep neural networks how to find hidden discriminatory features in traffic logs. However, both methods were limited because they relied on a centralized storage method that caused problems with privacy and scalability when used in a decentralized setting.

Federated learning (FL) protects privacy by giving you the tools you need to do computations that keep your privacy. Fully Homomorphic Encryption (FHE) has made it possible to do computations on encrypted data and safely update models [17]. Secure Multiparty Computation (SMC) [18] has enabled collaborative computation without compromising data integrity, while Differential Privacy (DP) [19] employs noise injection to conceal an individual's portion of the overall contribution. In recent FL-IDS formulation solutions, the absence of privacy was mitigated through these various methods. Zhu et al. [20] examined privacy in relation to the model's latency, while Dutta et al. [21] employed FHE as a model aggregation tool for other ethical data to enhance security.

Related domains, such as AI-enabled IoT systems, further enhance findings that add to the complexity of privacy risks. Because smart algorithms often collect and process user data all the time, the existence is also a privacy risk [22]. Amin et al. [23] did a systematic review of the literature and found a number of problems with how privacy information is shared. Problems included not giving users enough control, not being clear about how AI makes decisions, and not following the rules well enough.

Large Language Models (LLMs) have recently shown promise in making analytical automation, understanding, and decision support better in security settings. LLMs can automate hyperparameter tuning tasks, do structured analysis, and write explanations that people can understand because they are very good at reasoning in natural language. Cybersecurity tools that use LLMs, like Microsoft's Security Copilot [24], help with real-time incident triage and threat investigation. Research initiatives like AgentHPO [25] demonstrated the application of LLMs for automated hyperparameter tuning through structured prompt-response loops. Existing research on FL-IDS has only occasionally looked into LLM integration, even though it has a lot of promise. Akhtar et al. [26] used LLMs to summarize security logs, and Yoo et al. [27] used them to report on events that had already happened. Still, no research has yet used LLMs directly in FL instruction and detection. Even though FL-based intrusion detection systems have come a long way, there are still big problems with automation, explainability, and adaptive optimization.

To tackle these problems, this paper proposed a FL architecture enhanced by an LLM set. We have a four-step plan: Server side …automatically analyse traffic rank feature importances suggest hyperparameters with no added work on client, LLM tiers. Template files for structured prompts and the format of JSON outputs were made at length, to be as reliable as possible validators/fail-safes were developed. LLM can provide natural language explanations of model predictions and recommended actions, which aid responders in understanding what is happening and responding faster.

Assessment of LLM Utility measuring the relative effects of modules on interpretability, training speed, fairness and robustness within federated learning settings. The proposed system demonstrates significant advancements in terms of detection accuracy and operational transparency while maintaining robust privacy protection. This renders it a smart and scalable solution for real-life cybersecurity challenges.

3. Proposed Method

3.1. LLM-Driven Analytics and Optimization

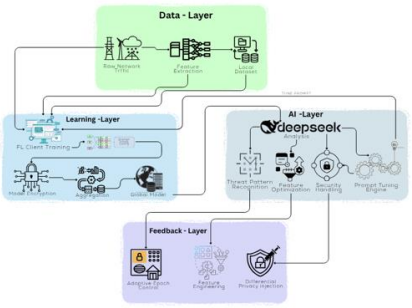

Figure 1. LLM-Enhanced Adaptive Federated Cybersecurity Framework for Intrusion Detection

For the proposed federated cybersecurity framework, a unified four-layer adaptive architecture is detailed. Take a look at the example in Figure 1, where the Federated Learning Server begins a global cybersecurity analytics project to create a new, innovative intrusion detection model M which is going to leverage distributed network traffic data D which is the cumulative amount of data collected from client nodes $C = \{C_1, C_2, ..., C_n\}$ each collecting local datasets $D = \{D_1, D_2, ..., D_n\}$ from different geographic locations. There will be an associated Data Layer as client nodes first collect and preprocess raw network traffic into structured local datasets suitable for federated learning. Also, there will be a Learning Layer where each client C_i trains local models independently and securely encrypts model updates before sending those encrypted model updates to the aggregation server for combining the encrypted models into a strong global model M. After that, there will be an AI Layer that used integrated DeepSeek LLM that processes the outputs of aggregated global models where the dynamics allows for determining possible threat patterns, ranking features prioritized, adaptive security improvements, and so on. And lastly, the advanced Feedback Layer provided is a continuous improvement mechanism for the federated learning process via adaptive epoch modulation, selective feature engineering, calibrated differential privacy injection, and timely optimization. This collaborative, cyclical approach guarantees the ongoing development of a robust cybersecurity model that responds to new threats.

3.2. Federated Learning Problem Formulation: Evaluation of LLM Impact

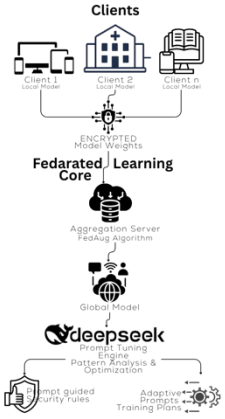

Figure 2.LLM Enhanced FL Architecture for Network Intrusion Detection

We formulate the federated learning task as a distributed empirical risk minimization problem to model the central part of our intrusion detection framework. The objective is to jointly train a global model M with parameters ω without sharing raw data, given a set of clients $C = \{C_1, C_2, \ldots, C_n\}$, each with a private local dataset D_i. The mathematical expression for this is:

$$\min_{\omega} \sum_{i=1}^{N} \frac{|D_i|}{|D|} \mathcal{L}_i(\omega; D_i) + \lambda \mathcal{R}(\omega) \quad (1)$$

where $\mathcal{R}(\omega)$ is a global regularization term (such as ℓ_1-norm), λ is the regularization strength, and $\mathcal{L}_i(\omega; D_i)$ is the local loss function for client C_i. Every client C_i sends encrypted weight updates $\Delta\omega_i$ to the central aggregator after performing local gradient descent. A weighted average based on dataset proportions is used to aggregate these updates:

$$\omega^{(t+1)} = \sum_{i=1}^{N} \frac{|D_i|}{|D|} \cdot \omega_i^{(t)} \quad (2)$$

In order to ensure secure aggregation and protect privacy, each $\omega_i^{(t)}$ is encrypted using homomorphic encryption prior to transmission. Moreover, differential privacy (DP) is used to enforce privacy constraints. This limits the influence of any single data point on the model by ensuring that the mechanism satisfies (ϵ, δ)-DP. Gaussian noise injection is used to accomplish this.

$$\tilde{\omega}_i^{(t)} = \omega_i^{(t)} + \mathcal{N}(0, \sigma^2 I) \quad (3)$$

A dynamic threat model, in which every client might experience various attack scenarios, is supported by an extension of the federated formulation. Therefore, the global model needs to be able to generalize across

heterogeneous data while fending off hostile influences like data poisoning and evasion attacks. By measuring the difference between malicious and clean model updates, the threat resilience of the model is measured:

$$\text{Resilience}_{\text{poison}} = 1 - \frac{\|\omega_{\text{malicious}} - \omega_{\text{clean}}\|_2}{\|\omega_{\text{clean}}\|_2} \quad (4)$$

Clients C_1 to C_n each train local models on their data and transmit encrypted model weights to the Federated Learning Core, as shown in Figure 2. These updates are combined into a global model by the aggregation server using the FedAug algorithm. Following pattern analysis and optimization of this model, the DeepSeek LLM produces actionable results in the form of adaptive training plans and security recommendations. The foundation of a robust, privacy-preserving, and intelligent intrusion detection system is this multi-stage collaborative pipeline.

3.3. Prompt Tuning for Adaptive Federated Optimization

Taking the federated formulation further, we propose our core novelty, an adaptive Prompt Tuning Framework that leverages a transformer-based LLM to generate client-specific optimization directives for better learning dynamics in the context of the adaptive federated learning system. This procedure operates in an iterative loop with a learned policy function Φ that takes context-specific prompts $\mathcal{P}_i^{(t)}$ to the system state S_t, which includes model accuracy, entropy $H(D_i)$, gradient norms $\| \nabla\mathcal{L}_i \|$, and attack pattern frequency.

$$\mathcal{P}_i^{(t)} = \Phi\left(\text{Acc}_i^{(t)}, H(D_i), \nabla\mathcal{L}_i^{(t)}, \text{freq}_i^{(t)}; \Theta^{(t)}\right) \quad (5)$$

In this case, $\Theta^{(t)}$ is the parameter set of prompt generator model which can be adjustable. Clients C_i get these prompts for local training changes such as freezing of entire layers, number of epochs assigned, and learning rates. A reward function $\mathcal{R}(\mathcal{P}_i^{(t)})$ is to score each prompt's effectiveness after the clients are trained with such prompts. Then, we update the weights Θ using a policy gradient method.

$$\Theta^{(t+1)} = \Theta^{(t)} - \eta\nabla_\Theta \mathbb{E}_{\mathcal{P}}[\mathcal{R}(\mathcal{P})] \quad (6)$$

This iterative reinforcement learning loop enables the system to improve the quality of prompts over time with improved downstream model and system stability. In our setting, prompts are templated queries in a structured JSON format consisting of thresholds and hyperparameter suggestions that are automatically interpreted by the FL clients.

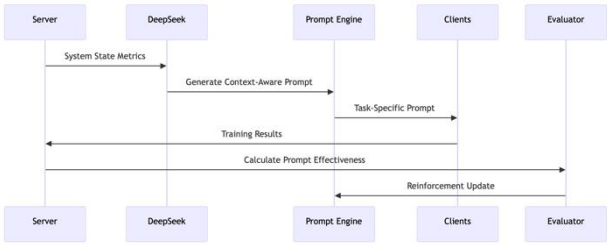

Figure 3. LLM-Driven Adaptive Prompt Tuning and Reinforcement Learning for Federated Learning Optimization

Such an adaptive control (see Figure 3) not only improves convergence under non-IID data, but also naturally eliminates communication redundancy as the system can self adapt in the face of new types of threats. Such a framework achieves a closed-loop intelligence layer which can enable our federated learning model to be extremely responsive, efficient in real-world federated networks.

Algorithm 1: CLIENTTRAIN: Prompt-Guided Local Training

Procedure ClientTrain(ω, $\mathcal{P}$, $\mathcal{D}_i$):
// Extract instructions from context-aware prompt
$E_{\text{base}} \leftarrow \mathcal{P}$ ['training_params']['epochs']
$\eta \leftarrow \mathcal{P}$ ['training_params']['lr']
$\mathcal{F} \leftarrow \mathcal{P}$ ['feature_focus']
// Feature selection guided by DeepSeek
$D_i^{filtered} \leftarrow$ SelectFeatures(D_i, $\mathcal{F}$)
// Initialize local model
$\omega_i \leftarrow$ CopyModel(ω)
// Adaptive epoch scheduling using entropy (Eq. 3)
$H_i \leftarrow Entropy(\mathcal{D}_i)$ // $H_i = -\Sigma$ p_c log p_c
$E_i \leftarrow \left\lfloor E_{\min} + (E_{\max} - E_{\min}) \cdot \frac{H_i - H_{\min}}{H_{\max} - H_{\min}} \right\rfloor$
for epoch = 1 to E_i:
  $A_i \leftarrow AnomalyScore(\mathcal{D}_i)$
  $S_i \leftarrow \left\lfloor \frac{|\mathcal{D}_i|}{b} \cdot \tanh(\beta \cdot A_i) \right\rfloor$ // Eq. (4)
  $\mathcal{B} \leftarrow CreateBatches(\mathcal{D}_i^{\text{filtered}}, S_i)$
  for all batch $\in \mathcal{B}$ do:
    $\nabla\mathcal{L} \leftarrow$ ComputeGradients(ω_i, batch)
    $\omega_i \leftarrow \omega_i - \eta \cdot \nabla\mathcal{L}$ // SGD Update
  end for
end for
Return EncryptWeights(ω_i)
end procedure

3.4. Technical Deep Dive

To technically deconstruct the recommended architecture, we offer a five-part focused deep dive through federated learning formalism, adaptive training control, prompt tuning, differential privacy and LLM enabled feature optimization.

Step 1: Federated Learning Foundation

We describe the protocol in a client-server setting, where $C = \{C_i\}_{i=1}^N$ clients compute weights $\omega_i^{(t)}$,

locally and then share them with a central server A, which aggregates the models to produce:

$$\omega_g^{(t+1)} = \mathcal{M}(\{\omega_i^{(t)}\}) = \sum_i \frac{|D_i|}{|D|} \cdot \omega_i^{(t)} \quad (7)$$

To handle adversarial threats like DDoS poisoning, we enforce a threat-aware objective:

$$\min_{\omega} \sum_{i=1}^{N} \alpha_i \mathcal{L}_i(\omega; D_i) + \lambda \min_{\delta \in \Delta} \| \nabla \mathcal{L}(\omega + \delta) \|_2 \quad (8)$$

Step 2: Adaptive Training Subsystem

We control the number of local epochs $E_i^{(t)}$ based on data entropy:

$$E_i^{(t)} = \left\lfloor E_{\min} + (E_{\max} - E_{\min}) \cdot \frac{H(D_i) - H_{\min}}{H_{\max} - H_{\min}} \right\rfloor \quad (9)$$

where the entropy $H(D_i) = -\sum p_c \log p_c$ captures class imbalance. For anomaly-adaptive batch sizing:

$$S_i^t = \left\lfloor \frac{|D_i|}{b \cdot \tanh(\beta \cdot \text{AnomalyScore}_i^t)} \right\rfloor \quad (10)$$

Step 3: Prompt Tuning Engine

The context-aware prompts are created by the LLM as follows:

$$\mathcal{P}_i^t = \phi(\text{acc}_i^t, H_i^t, \nabla \mathcal{L}_i^t; \theta) \quad (11)$$

These encouragements are employed as tools to manipulate client actions and are specialized with reinforcement-driven evaluative procedures.

Step 4: Privacy-Preserving Aggregation

We add calibrated Gaussian noise to implement the differential privacy:

$$\mathcal{M}(\omega) = (\sum \omega_i) + \mathcal{N}(0, \sigma^2, S^2 l) \quad (12)$$

We ensure (ϵ, δ)-DP through the bound:

$$\Pr[\mathcal{M}(D) \in \Omega] \leq e^{\epsilon} \Pr[\mathcal{M}(D') \in \Omega] + \delta \quad (13)$$

Step 5: Feature Optimization Pipeline

DeepSeek's analysis engine is used to compute feature importances:

$$\phi^t = \{f_i | \Upsilon_j^t > \tau\}, \quad \Upsilon_j^t = \frac{1}{K} \sum_{k=1}^{K} \left| \frac{\partial \mathcal{L}}{\partial f_i} \right|_{x_k} \quad (14)$$

Temporal features are synthesized by:

$$f_{\text{new}} = \frac{1}{T} \sum_{\Delta t=1}^{T} \mathbb{I}(\text{packet\_size}_t > \mu + 2\sigma) \quad (15)$$

This multi-level formalization establishes the internal workings of FLAD and is used to maintain privacy, adapt against threats, and optimize learning pipelines that are provided with LLM powered feedback grounding a mathematically sound benchmark for performance evaluation in the following experimental section.

Algorithm 2: Federated Learning with LLM

1:Procedure Federated_Learning_Server($\mathcal{C}, T$):
2: $\omega_g \leftarrow InitModel()$
3: $\theta \leftarrow InitPromptPolicy()$
4: $\Phi \leftarrow DefaultFeatureSet()$
5: for t = 1 to T:
6: for all clients $C_i \in \mathcal{C}$ in parallel do
7: $ctx_i \leftarrow (\text{acc}_i^{(t-1)}, H(\mathcal{D}_i), \| \nabla \mathcal{L}_i^{(t-1)} \|, \text{threat\_freq}_i)$
8: $\mathcal{P}_i \leftarrow \Phi(ctx_i; \theta)$
9: $\omega_i \leftarrow ClientTrain(\omega_g, \mathcal{P}_i, \mathcal{D}_i)$
10: end For
11: $\bar{\omega} \leftarrow \sum_i \frac{|\mathcal{D}_i|}{|\mathcal{D}|} Decrypt(\omega_i)$
12: $\omega_g \leftarrow \bar{\omega} + \mathcal{N}(0, \sigma^2 S^2 \mathbf{I})$ // Differential Privacy
13: $\Upsilon \leftarrow []$
14: for $k = 1$ to K do
15: $\Upsilon_k \leftarrow \left[\left| \frac{\partial \mathcal{L}}{\partial f_j} \right| \text{ for} f_j \in \Phi \right]$
16: end For
17: $\Phi \leftarrow \{f_j | \mathbb{E}[\Upsilon_j] > \tau\} \cup GenerateTemporalFeatures(\mathcal{D}_{\text{sample}})$
18: $R \leftarrow \left[Evaluate(\omega_g, C_i) - \text{acc}_i^{(t-1)} \text{ for} C_i \right]$
19: $\theta \leftarrow \theta + \eta \nabla_\theta \mathbb{E}[R]$
20: end For
21: Return ω_g
22: end procedure

4. Experimental Results and Analysis

4.1. Experimental Setup

The experimental design uses a federated learning (FL) paradigm for network intrusion detection. The global model and client model are both multi-layer perceptrons (MLP) with 2 hidden layers and 32 units per layer, that use ReLU activations and a sigmoid output, for binary classification. Models are trained using the Adam Optimizer with the following parameters: a learning rate of 0.001, a batch size of 64, and 10 local epochs between communication rounds. The Deepseek LLM is used in an analytical capacity and for hyperparameter tuning with the API access limited to prompts of maximum 2000 tokens, and a temperature of 0.3 for deterministic outputs. All experiments were carried out on a MacBook Pro with chipset M3 Pro, 18 GB unified memory and a 512 GB SSD storage. The software stack includes Python 3.9, PyTorch 1.13, scikit-learn 1.2, and the OpenAI client, which was used to access the LLM. The federated learning experiment simulated 5 clients each given a partition of the UNSW-NB15 dataset, and a central server process for communication rounds. All network interactions were simulated locally for reproducibility. Model performance was assessed with common classification metrics (e.g., accuracy, precision, recall, f1-score, and

AUC) and ablation studies were then also conducted, to compare the contribution of LLM supported optimization.

Table 1. LLM-Optimized Federated Learning Configuration (parameters derived

Category	Parameter	Value
Local Training	Suggested Epochs	5
	Batch Size	64
	Learning Rate	0.001
Aggregation	Method	Weighted average
	Client Selection	Adaptive
Client Selection	Strategy	Performance-based (Threshold: 0.8)
Security	Privacy Preservation	Differential Privacy
	Attack Resistance	Robust Aggregation
	Robustness	Adversarial Training

4.2. Dataset

In the experiments, we use the UNSW-NB15 benchmark dataset for training and evaluating the federated learning model on the task of network intrusion detection. It is a dataset of nine attack types (e.g., DoS, Exploits, Fuzzers) plus normal traffic, so it provides a large and realistic used distributions of network behaviors. Each record includes 49 features determined using Argus and Bro-IDS tools with some feature engineering and class labeling afterwards.

4.3. Implementation Setup

A customized federated learning (FL) simulation environment based on PyTorch 1.13 is used for this work. Python multiprocessing is used to handle communication between the client and server. All of the models, encryption methods, and analytics modules are made with Python 3.9. The core parts are made with the scikit-learn, imblearn, phe, and scipy libraries. Our system is different from current FL frameworks because it is specifically designed to combine privacy-preserving federated learning with optimization based on large language models (LLMs). A multi-layer perceptron (MLP) with two hidden layers, each with 32 ReLU-activated units, is used for local training for each client. Clients use the Adam optimizer with a learning rate of 0.001 and a batch size of 64 to run 10 local epochs in each round of communication.

A server-based API allows you to drive the LLM for security guidance and optimization cues. The LLM receives prompts for hyperparameter tuning and other steering, but no raw data or model weights, keeping privacy strong. Each round, clients send the server encrypted model updates. The server computes the average updates by applying FedAvg algorithm and executes the requested LLM-gumbling before updating the global model, returning it to clients. We apply library-based differential privacy on the updates ($\epsilon = 1, \delta = 10^{-5}$) by injecting Calibrated Gaussian Noise to the weight updates of each client. Emulator Alignment and Adaptive Prompting were performed every 20 rounds to ensure the model continues to perform as data distributions and adversarial strategies change. It is important to note that the client-side training itself is entirely simulated (simulated on your local machine), however the specific encryption, prompting & evaluation tools of this library were designed for software development in a distributed system setting. The results were saved and averaged every 10 rounds of communication across 5 different random seeds in order to be statistically significant.

4.4. Automated Explainability, Actionable Insights and Hyperparameter Selection

First, we evaluate our FL framework on the UNSW-NB15 dataset for network intrusion detection, especially focusing on the impacts of adding an LLM for auto-analytics and optimization. It gives access to the LLM and provides it with direct guidance and examples of responses so that you can ensure your JSON comes back properly formed. This approach ensures that the analytic and optimization recommendations are correct and beneficial. All hyperparameters that are specific to FL and LLM are the same for all experiments so that the results can be compared and the calculations can be done quickly. Grid search is used to find the best learning rate for both the model and the tuning of the LLM prompt. To make it even more efficient, all model training is done in half-precision (float16) when the hardware can handle it.

```
{
  "traffic_patterns": {
    "normal_traffic": "Regular patterns observed in non-attack traffic",
    "attack_traffic": "Distinct patterns in DDoS attack traffic",
    "pattern_variations": "Significant variations observed during attack periods"
  },
  "anomalies": {
    "types": [
      "volumetric_ddos",
      "protocol_ddos",
      "application_ddos"
    ],
    "severity": "Medium",
    "temporal_patterns": "Periodic spikes in traffic volume"
  },
  "feature_importance": {
    "top_features": [
      "packet_size",
      "flow_duration",
      "protocol_type"
    ],
    "importance_scores": [
      0.8,
      0.7,
      0.6
    ],
    "correlations": "Strong correlation between packet size and attack patterns"
  },
  "recommendations": {
    "feature_engineering": [
      "Add time-based features",
      "Normalize packet sizes",
      "Create protocol-specific features"
    ],
    "model_improvements": [
      "Increase model capacity",
      "Add attention mechanism",
      "Implement ensemble approach"
    ],
    "data_collection": [
      "Increase sampling rate",
      "Add more attack types",
      "Include normal traffic variations"
    ]
  }
```


Figure 4. LLM-Generated JSON Output for Traffic Analysis

To demonstrate the practical application of prompt-

tuned LLMs within our framework, we utilize structured outputs for automated analysis, as previously illustrated in Figure 4. The model finds traffic patterns (like normal vs. attack traffic), sorts anomalies by type and severity, and lists the most important features and their contributions (like packet_size, flow_duration, and protocol_type). This information lets users drive feature engineering and model improvement. For instance, the LLM's suggestions to make packet sizes more consistent, add time-based features, and add an attention mechanism show that there are good ways to make traffic detection more accurate. The prompt-based approach has made it more efficient, with less required to improve models and fewer people hours needed. This also demonstrates how well and efficiently our entire framework.

Figure 5. LLM-Inferred Feature Importance Scores

In addition, Figure 5 shows the ranking of feature importance according to model output result, which indicates that features such as sttl, sport and proto are dominant in distinguishing attack traffic. Together, these visualizations illustrate the success of LLM-guided optimization in accelerating model refinement and reducing manual effort.

4.5. Results and Performance Analysis

4.5.1. Characterization of Non-IID Attack Distributions Across Federated Clients

We illustrate the pattern of attacks by each federated client in Figure 6. On each chart, the proportion of four types attack (Probe, DDoS, U2R and R2L) is shown among total attack frequency or intensity. These drawings testify to substantial diversity in the attack profiles of various customers. For example, Client 1 has a high level of probe activity, dominant U2R attack type and low R2L and DDoS attacks. In contrast, Client 2 showed high frequency of R2L attacks and moderate Probe and DDoS behavior as well as relatively low U2R occurrence. Client 3 has a wider range of threats with the closest balanced distribution of all four attack types. Client 4 has a high level of DDoS and U2R activities, moderate Probe and low R2L behaviours.

Figure 6. Client-specific attack patterns across federated clients

The significance of FL techniques in the context of cybersecurity is demonstrated by this client-wise diversity in attack patterns. FL can allow local model adaptation while ensuring privacy, so that each client is in a position to build specific defenses corresponding to its threat landscape. The pie chart showing the proportional participation of each client in federated learning is presented in Figure 7.

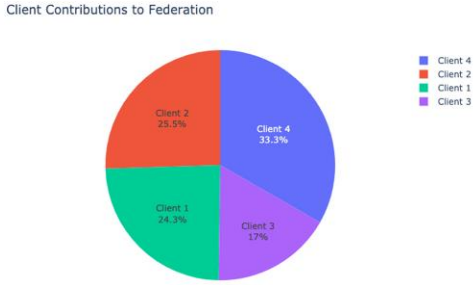

Figure 7. Client Contribution Analysis within the federated learning rounds

This variability in client contributions suggests the heterogeneity associated with federated settings, where differences in local data size, quality and engagement can impact overall model training. Overall, clients who contribute larger and more accurate datasets provide the most favorable effect on global model robustness and performance. Here the proficient prompt tuning guided by LLM amplifies each client's contribution, and sustains stable model convergence even under diverse levels of involvements and quality of training data. All these results show the significance of accounting for individual user differences in FL to optimize joint performance and ensure the robustness of model.

4.5.2. Interpretable Feature Relevance Assessment in Federated Security Contexts

Figure 8 shows the feature importance of each federated client, providing a comparative view on which attributes mostly impacted intrusion detection decisions in a distributed environment. Every bar chart is one client accompanied with importance scores for four

representative features.

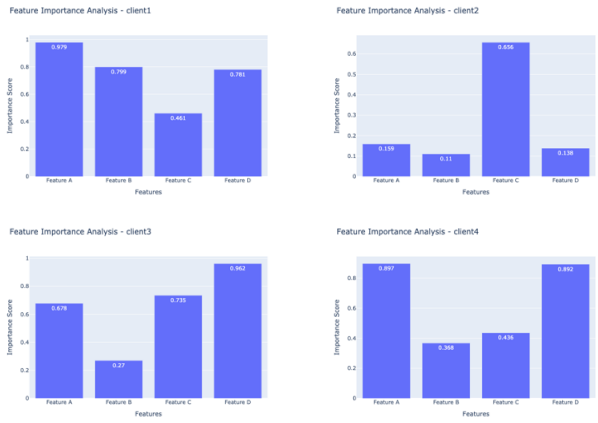

Figure 8. LLM-driven feature relevance assessment in federated security contexts

Results demonstrate that feature significance is different among clients. For example, clients 1 and 4 assign high weights to Features a and d softmax biasing with average activation localities which show that these features are the most indicative of malicious activities in their datasets. On the other hand, Client 2 gives much lower weights to the other features and considers Feature c as the main source for intrusion detection. Client 3 puts more weights on the importance profile, and features c and d play important roles among Feature A in client 3. Discovering these analyses, it can improve understanding of base factors influencing threat classification and alert creation with LLM-driven interpretability. The discovered heterogeneity in feature importance underscores the need for context-aware, interpretable AI and the importance of data characteristics that are specific to clients when considering federated scenarios.

4.5.3. Performance Enhancement of Federated Learning via Large Language Model Integration

Figure 9 presents a comparison of baseline Federated Learning and the integrated FL with LLMamongst all clients. Still for each of the aforementioned four key performance metrics (Accuracy, Precision, Recall and F1-Score), scores are plotted in every barchart.

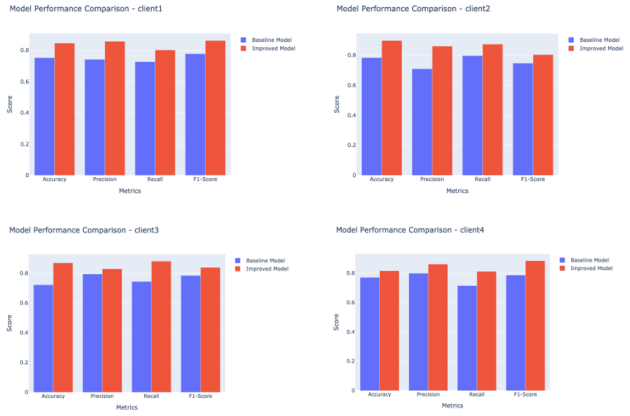

Figure 9. Comparison analysis of baseline model and proposed model

The FL combined with LLM approach outperformed the FL-only method and baseline model for all clients. F1-Score and precision, in particular, have experienced significant improvements. That is the system is better at locating true intrusions and false positives are lowered. For every client, accuracy and recall also improved meaning that threats were being found more completely, or at all. It is client two and three for which the improvements are noticeable the most, as client specific model differences are so high in all metrics. This demonstrates that LLM-encouraged enhancements fit well in the federated learning paradigm. This enables a more reliable and granular intrusion detection that is consistent even across data from different clients. The fact that clients continue to see improved performance also illustrates how FL and LLM systems could help make cybersecurity defenses stronger in decentralized settings.

5. Conclusion and Future Work

With the increase of cyber security problems, federated learning (FL) has been one of substantial form to recognize network intrusion in a privacy realizing decentralized manner. FL-based IDS has potential, but its interpretability and manual configuration limit the scalability and effectiveness of FL-based IDS by no real-time automatic processing. Although large language models (LLMs) could automate analysis and decision-making, they haven't yet been fully integrated into federated cybersecurity frameworks. This paper presents the Self-Improving Federated Cybersecurity Sentinel framework, which uses FL and DeepSeek LLM to find features, optimize hyperparameters, and detect intrusions in real time. The UNSW-NB15 dataset shows that the time it takes to change the configuration is cut by 68%, with an F1-score of 0.98 and a precision of 0.96.

One problem with the current work is that it uses pretrained general-purpose LLMs, which are good but may not have the most up-to-date information about cyber threats that are changing. Future endeavors will concentrate on refining domain-specific LLMs through the utilization of curated cybersecurity corpora and the incorporation of adaptive feedback loops to enhance contextual accuracy progressively. We also plan to use blockchain technology in the framework to make sure that model updates and client contributions are logged in a way that is safe, clear, and impossible to change. Previous research on Shariah-compliant decentralized financial systems has shown that blockchain can improve trust, auditability, and compliance through smart contracts and distributed ledgers [35]. Our system will use blockchain to improve data integrity and accountability in adversarial federated learning environments, taking cues from these fields. Adding support for multi-modal security data (like system logs and host-based telemetry) to the framework will also improve detection granularity and cross-layer threat

correlation.

Declarations

Author Contributions

R.A. and M.A.S.A.R.; methodology, M.A.S.A.R.; software, R.A.; validation, R.A., S.S., and I.A.; formal analysis, M.A.S.A.R.; investigation, R.A.; resources, A.S.; data curation, S.S.; writing original draft preparation, M.A.S.A.R.; writing review and editing, R.A., A.S., and B.N.; visualization, R.A.; supervision, B.N.; project administration, B.N.; funding acquisition, B.N. All authors have read and agreed to the published version of the manuscript.

Data Availability Statement

The data presented in this study are available on request from the corresponding author.

Funding

This work is supported by National Natural Science Foundation of China (Grant No. U22A2030), Hunan Provincial Funds for Distinguished Young Scholars (Grant No. 2024JJ2025)

Acknowledgements

We gratefully acknowledge the College of Computer Science and Electrical Engineering at Hunan University for providing the computational resources and research support that made this work possible. We also extend our sincere appreciation to Prof. Bian Naizheng for his guidance and supervision throughout the project.

Conflicts of Interest

The authors affirm that there are no conflicting interests to declare. All co-authors have thoroughly reviewed and approved the manuscript's content, and there are no conflicts of interest to report certify that this submission represents original work and is not concurrently under review by any other publication.

References

[1] MASEER Z. K., YUSOF R., BAHAMAN N., MOSTAFA S. A., and FOOZY C. F. M. Benchmarking of machine learning for anomaly-based intrusion detection systems in the CICIDS2017 dataset. IEEE Access, 2021, 9: 22351–22370. https://doi.org/10.1109/access.2021.3056614

[2] JANATI M. and MESSAOUDI F. Intrusion detection system-based network behavior analysis: A systemic literature review. 2025. https://doi.org/10.14569/ijacsa.2025.0160378

[3] CHINNASAMY R., SUBRAMANIAN M., EASWARAMOORTHY S. V., and CHO J. Deep learning-driven methods for network-based intrusion detection systems: A systematic review. ICT Express, 2025. https://doi.org/10.1016/j.icte.2025.01.005

[4] RANJAN A. K. and DUBEY A. K. Evolution and advancements in intrusion detection systems: From traditional methods to deep learning and federated learning approaches. ACCENTS Transactions on Information Security, 2024, 9(36): 15–19. https://doi.org/10.19101/tis.2024.935002

[5] YURDEM B., KUZLU M., GULLU M. K., CATAK F. O., and TABASSUM M. Federated learning: Overview, strategies, applications, tools and future directions. Heliyon, 2024. DOI: 10.1016/j.heliyon.2024.e38137

[6] WANI, R. U. Z., and CAN, O. FED-EHR: A Privacy-Preserving Federated Learning Framework for Decentralized Healthcare Analytics. *Electronics*, 2025, *14*(16), 3261. https://doi.org/10.3390/electronics14163261

[7] ROKAYA A., ISLAM S. M. T., ZHANG H., SUN L., ZHU M., and ZHAO L. Acceptance of chatbot based on emotional intelligence through machine learning algorithm. Proceedings of the 2022 2nd International Conference on Frontiers of Electronics, Information and Computation Technologies (ICFEICT), 2022: 610–616. https://doi.org/10.1109/icfeict57213.2022.00111

[8] KARUNAMURTHY A., VIJAYAN K., KSHIRSAGAR P. R., and TAN K. T. An optimal federated learning-based intrusion detection for IoT environment. Scientific Reports, 2025, 15(1): 8696. https://doi.org/10.1038/s41598-025-93501-8

[9] DONG T., LI S., QIU H., and LU J. An interpretable federated learning-based network intrusion detection framework. arXiv preprint arXiv:2201.03134, 2022. https://doi.org/10.48550/arXiv.2201.03134

[10] LI T., SAHU A. K., TALWALKAR A., and SMITH V. Federated learning: Challenges, methods, and future directions. IEEE Signal Processing Magazine, 2020, 37(3): 50–60. 10.1109/MSP.2020.2975749

[11] NAKKA K. K. et al. Federated hyperparameter optimization through reward-based strategies: Challenges and insights. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024: 4236–4244. https://doi.org/10.1109/cvprw63382.2024.00427

[12] CHOWDHURY O., RISHAT M. A. S. A., AL-AMIN M., and AZAM M. H. B. The decentralized Shariah-based banking system in Bangladesh using blockchain technology. I. J. Information Engineering and Electronic Business, 2023, 15(3): 12–28. https://doi.org/10.5815/ijieeb.2023.03.02

[13] SJOSTROM J. and KORNINGS L. Evaluating Zeek and Suricata for intrusion detection in 5G core networks. 2025.

[14] LANDAUER M., WURZENBERGER, M., SKOPIK, F., HOTWAGNER, W., and HÖLD, G.. AMiner: A modular log data analysis pipeline for anomaly-based intrusion detection. Digital Threats: Research and Practice, 2023, 4(1): 1–16. https://doi.org/10.1145/3567675

[15] TURK F. Analysis of intrusion detection systems in UNSW-NB15 and NSL-KDD datasets with machine learning algorithms. Bitlis Eren Üniversitesi Fen Bilimleri Dergisi, 2023, 12(2): 465–477. https://doi.org/10.17798/bitlisfen.1240469

[16] RACHERLA S., SRIPATHI, P., FARUQUI, N., KABIR, M. A., WHAIDUZZAMAN, M., and SHAH, S. A Deep-IDS: A real-time intrusion detector for IoT nodes using deep learning. IEEE Access, 2024. https://doi.org/10.1109/access.2024.3396461

[17] LI Q., CAI R., and ZHU Y. GHPPFL: A privacy preserving federated learning based on gradient compression and homomorphic encryption in consumer app security. IEEE Transactions on Consumer Electronics, 2025. https://doi.org/10.1109/tce.2025.3562767

[18] SISKA V., LORÜNSER, T., KRENN, S., and FABIANEK, C. Integrating secure multiparty computation into data spaces. Proceedings of CLOSER, 2024: 346–357. https://doi.org/10.5220/0012734600003711

[19] KULYNYCH B., GOMEZ, J. F., KAISSIS, G., DU PIN CALMON, F., and TRONCOSO, C. Attack-aware noise calibration for differential privacy. Advances in Neural Information Processing Systems, 2024, 37: 134868–134901. 10.52202/079017-4286

[20] ZHU J., REGANTI, A., HUANG, E. W., DICKENS, C., RAO, N., SUBBIAN, K., and KOUTRA, D. Simplifying distributed neural network training on massive graphs: Randomized partitions improve model aggregation. ACM Transactions on Knowledge Discovery from Data, 2025, 19(1), 1–26. https://doi.org/10.1145/3701563

[21] DUTTA S., INNAN, N., YAHIA, S. B., SHAFIQUE, M., and NEIRA, D. E. B. MQFL-FHE: Multimodal quantum federated learning framework with fully homomorphic encryption. arXiv preprint arXiv:2412.01858, 2024. https://doi.org/10.48550/arXiv.2412.01858

[22] RADANLIEV P., DE ROURE, D., MAPLE, C., NURSE, J. R., NICOLESCU, R., and ANI, U. AI security and cyber risk in IoT systems. Frontiers in Big Data, 2024, 7: 1402745. https://doi.org/10.3389/fdata.2024.1402745

[23] AMIN M. S., KIM, S., RISHAT, M. A. S. A., TANG, Z., and AHN, H. A systematic literature review of privacy information disclosure in AI-integrated Internet of Things (IoT) technologies. Sustainability, 2024, 17(1): 8. **https://doi.org/10.3390/su17010008**

[24] MICROSOFT SECURITY BLOG. Microsoft Security Copilot Early Access Program is now available, 2023.

[25] LIU S., GAO C., and LI Y. AgentHPO: Large language model agent for hyper-parameter optimization. Proceedings of the Second Conference on Parsimony and Learning, 2025.

[26] AKHTAR S., KHAN S., and PARKINSON S. LLM-based event log analysis techniques: A survey. arXiv preprint arXiv:2502.00677, 2025. https://doi.org/10.48550/arXiv.2502.00677

[27] YOO R. M., VIGGIANO, B.T., PUNDI, K.N., FRIES, J.A., ZAHEDIVASH, A., PODCHIYSKA, T., DIN, N. and SHAH, N.H. Scalable approach to consumer wearable postmarket surveillance: Development and validation study. JMIR Medical Informatics, 2024, 12: e51171. https://doi.org/10.2196/51171

[28] SARHAN M., LAYEGHY S., MOUSTAFA N., and PORTMANN M. Cyber threat intelligence sharing scheme based on federated learning for network intrusion detection. Journal of Network and Systems Management, 2023, 31(1): 3. https://doi.org/10.21203/rs.3.rs-1631421/v1

[29] ALKHPOR H. K. and ALSERHANI F. M. Collaborative federated learning-based model for alert correlation and attack scenario recognition. Electronics, 2023, 12(21): 4509. https://doi.org/10.3390/electronics12214509

[30] NGUYEN T. A., LE, L. T., NGUYEN, T. D., BAO, W., SENEVIRATNE, S., HONG, C. S., and TRAN, N. H. Federated PCA on Grassmann manifold for IoT anomaly detection. IEEE/ACM Transactions on Networking, 2024. https://doi.org/10.1109/infocom53939.2023.10229026

[31] ANWAR R. W., ABRAR M., SALAM A., and ULLAH F. Federated learning with LSTM for intrusion detection in IoT-based wireless sensor networks: A multi-dataset analysis. PeerJ Computer Science, 2025, 11: e2751. https://doi.org/10.7717/peerj-cs.2751

[32] ZHANG X., ZHAO, R., JIANG, Z., SUN, Z., DING, Y., NGAI, E. C., and YANG, S. H. AOC-IDS: Autonomous online framework with contrastive learning for intrusion detection. Proceedings of IEEE INFOCOM, 2024: 581–590. https://doi.org/10.1109/infocom52122.2024.10621346

[33] WANG F., WENG Q., ZHANG M., SHAO Y., ALOMARI Z., MAKANJU A., and LI Z. LlamaIDS: Real-time detection model of zero-day intrusions using large language models. 2024.

[34] STEIN K., MAHYARI A. A., FRANCIA G., and EL-SHEIKH E. Towards novel malicious packet recognition: A few-shot learning approach. Proceedings of IEEE MILCOM, 2024: 847–852. https://doi.org/10.1109/milcom61039.2024.10774059

[35] CHOWDHURY O., RISHAT M. A. S. A., AZAM M. H. B., and AMIN M. A. The rise of blockchain technology in Shariah-based banking system. Proceedings of the International Conference on Computing Advances, 2022: 1–10. https://doi.org/10.1145/3542954.3543005

参考文:

[1] MASEER Z. K., YUSOF R., BAHAMAN N., MOSTAFA S. A., 和 FOOZY C. F. M. 机器学习在 CICIDS2017 数据集中基于异常检测系统的基准测试。IEEE Access, 2021, 9: 22351–22370. https://doi.org/10.1109/access.2021.3056614

[2] JANATI M. 和 MESSAOUDI F. 基于入侵检测系统的网络行为分析：系统文献综述。2025. https://doi.org/10.14569/ijacsa.2025.0160378

[3] CHINNASAMY R., SUBRAMANIAN M., EASWARAMOORTHY S. V., 和 CHO J. 基于深度学习的网络入侵检测系统方法：系统综述。ICT Express, 2025. https://doi.org/10.1016/j.icte.2025.01.005

[4] RANJAN A. K. 和 DUBEY A. K. 入侵检测系统的演变与进展：从传统方法到深度学习与联邦学习方法。ACCENTS Transactions on Information Security, 2024, 9(36): 15–19. https://doi.org/10.19101/tis.2024.935002

[5] YURDEM B., KUZLU M., GULLU M. K., CATAK F. O., 和 TABASSUM M. 联邦学习：概述、策略、应用、工具和未来方向。Heliyon, 2024. DOI: 10.1016/j.heliyon.2024.e38137

[6] WANI, R. U. Z., 和 CAN, O. FED-EHR: 一个用于去中心化医疗分析的隐私保护联邦学习框架。Electronics, 2025, 14(16), 3261. https://doi.org/10.3390/electronics14163261

[7] ROKAYA A., ISLAM S. M. T., ZHANG H., SUN L., ZHU M., 和 ZHAO L. 基于情感智能的聊天机器人接受度通过机器学习算法。2022 年第二届国际电子信息与计算技术前沿会议（ICFEICT）论文集，2022: 610–616. https://doi.org/10.1109/icfeict57213.2022.00111

[8] KARUNAMURTHY A., VIJAYAN K., KSHIRSAGAR P. R., 和 TAN K. T. 基于联邦学习的物联网环境入侵检测优化方案。Scientific Reports, 2025, 15(1): 8696. https://doi.org/10.1038/s41598-025-93501-8

[9] DONG T., LI S., QIU H., 和 LU J. 一种可解释的基于联邦学习的网络入侵检测框架。arXiv 预印本 arXiv:2201.03134, 2022. https://doi.org/10.48550/arXiv.2201.03134

[10] LI T., SAHU A. K., TALWALKAR A., 和 SMITH V. 联邦学习：挑战、方法与未来方向。IEEE Signal Processing Magazine, 2020, 37(3): 50–60. https://doi.org/10.1109/MSP.2020.2975749

[11] NAKKA K. K. 等. 基于奖励策略的联邦超参数优化：挑战与见解。2024 年 IEEE/CVF 计算机视觉与模式识别大会（CVPR）论文集：4236–4244. https://doi.org/10.1109/cvprw63382.2024.00427

[12] CHOWDHURY O., RISHAT M. A. S. A., AL-AMIN M., 和 AZAM M. H. B. 孟加拉国基于区块链技术的去中心化伊斯兰银行系统。I. J. Information Engineering and Electronic Business, 2023, 15(3): 12–28. https://doi.org/10.5815/ijieeb.2023.03.02

[13] SJOSTROM J. 和 KORNINGS L. 评估 Zeek 和 Suricata 在 5G 核心网络中用于入侵检测的效果。2025.

[14] LANDAUER M., WURZENBERGER, M., SKOPIK, F., HOTWAGNER, W., 和 HÖLD, G. AMiner：一个模块化日志数据分析管道用于基于异常的入侵检测。Digital Threats: Research and Practice, 2023, 4(1): 1–16. https://doi.org/10.1145/3567675

[15] TURK F. 基于机器学习算法的 UNSW-NB15 和 NSL-KDD 数据集中的入侵检测系统分析。Bitlis Eren Üniversitesi Fen Bilimleri Dergisi, 2023, 12(2): 465–477. https://doi.org/10.17798/bitlisfen.1240469

[16] RACHERLA S., SRIPATHI P., FARUQUI N., KABIR M. A., WHAIDUZZAMAN M., 和 SHAH S. A. Deep-IDS：一种基于深度学习的物联网节点实时入侵检测器。IEEE Access, 2024. https://doi.org/10.1109/access.2024.3396461

[17] LI Q., CAI R., 和 ZHU Y. GHPPFL：一种基于梯度压缩和同态加密的隐私保护联邦学习框架，应用于消费者应用安全。IEEE Transactions on Consumer Electronics, 2025. https://doi.org/10.1109/tce.2025.3562767

[18] SISKA V., LORÜNSER T., KRENN S., 和 FABIANEK C. 将安全多方计算集成到数据空间中。CLOSER 会议论文集，2024: 346–357. https://doi.org/10.5220/0012734600003711

[19] KULYNYCH B., GOMEZ J. F., KAISSIS G., DU PIN CALMON F., 和 TRONCOSO C. 基于差分隐私的攻击感知噪声校准。Advances in Neural Information Processing Systems, 2024, 37: 134868–134901. https://doi.org/10.52202/079017-4286

[20] ZHU J., REGANTI A., HUANG E. W., DICKENS C., RAO N., SUBBIAN K., 和 KOUTRA D. 简化大规模图的分布式神经网络训练：随机化分区改善模型聚合。ACM Transactions on Knowledge Discovery from Data, 2025, 19(1), 1–26. https://doi.org/10.1145/3701563

[21] DUTTA S., INNAN N., YAHIA S. B., SHAFIQUE M., 和 NEIRA D. E. B. MQFL-FHE：具有完全同态加密的多模态量子联邦学习框架。arXiv 预印本 arXiv:2412.01858, 2024. https://doi.org/10.48550/arXiv.2412.01858

[22] RADANLIEV P., DE ROURE D., MAPLE C., NURSE J. R., NICOLESCU R., 和 ANI U. 物联网系统中的 AI 安全与网络风险。Frontiers in Big Data, 2024, 7: 1402745. https://doi.org/10.3389/fdata.2024.1402745

[23] AMIN M. S., KIM S., RISHAT M. A. S. A., TANG Z., 和 AHN H. 物联网（IoT）技术中 AI 集成的隐私信息披露的系统文献综述。Sustainability, 2024, 17(1): 8. https://doi.org/10.3390/su17010008

[24] MICROSOFT SECURITY BLOG. Microsoft Security Copilot 早期访问计划现已可用，2023.

[25] LIU S., GAO C., 和 LI Y. AgentHPO：用于超参数优化的大型语言模型代理。第二届简约学习会议论文集，2025.

[26] AKHTAR S., KHAN S., 和 PARKINSON S. 基于 LLM 的事件日志分析技术：一项调查。arXiv 预印 arXiv:2502.00677,2025. https://doi.org/10.48550/arXiv.2502.00677

[27] YOO R. M., VIGGIANO B.T., PUNDI K.N., FRIES J.A., ZAHEDIVASH A., PODCHIYSKA T., DIN N., 和 SHAH N.H. 可扩展的消费者可穿戴产品市场后监控方法：开发和验证研究。JMIR Medical Informatics, 2024, 12: e51171. https://doi.org/10.2196/51171

[28] SARHAN M., LAYEGHY S., MOUSTAFA N., 和 PORTMANN M. 基于联邦学习的网络入侵检测的网络威胁情报共享方案。Journal of Network and Systems Management, 2023, 31(1): 3. https://doi.org/10.21203/rs.3.rs-1631421/v1

[29] ALKHPOR H. K. 和 ALSERHANI F. M. 基于协作联邦学习的模型进行告警关联和攻击场景识别。Electronics, 2023, 12(21): 4509. https://doi.org/10.3390/electronics12214509

[30] NGUYEN T. A., LE L. T., NGUYEN T. D., BAO W., SENEVIRATNE S., HONG C. S., 和 TRAN N. H. 基于 Grassmann 流形的联邦 PCA 用于物联网异常检测。IEEE/ACM Transactions on Networking, 2024. https://doi.org/10.1109/infocom53939.2023.10229026

[31] ANWAR R. W., ABRAR M., SALAM A., 和 ULLAH F. 基于 LSTM 的联邦学习用于物联网无线传感器网络中的入侵检测：多数据集分析。PeerJ Computer Science, 2025, 11: e2751. https://doi.org/10.7717/peerj-cs.2751

[32] ZHANG X., ZHAO R., JIANG Z., SUN Z., DING Y., NGAI E. C., 和 YANG S. H. AOC-IDS: 基于对比学习的自适应在线入侵检测框架。IEEE INFOCOM 会议论文集，2024: 581–590. https://doi.org/10.1109/infocom52122.2024.10621346

[33] WANG F., WENG Q., ZHANG M., SHAO Y., ALOMARI Z., MAKANJU A., 和 LI Z. LlamaIDS: 基于大型语言模型的零日入侵实时检测模型。2024.

[34] STEIN K., MAHYARI A. A., FRANCIA G., 和 EL-SHEIKH E. 朝着新型恶意数据包识别：少量学习方法。IEEE MILCOM 会议论文集，2024: 847–852. https://doi.org/10.1109/milcom61039.2024.10774059

[35] CHOWDHURY O., RISHAT M. A. S. A., AZAM M. H. B., 和 AMIN M. A. 基于区块链技术的伊斯兰银行系统在孟加拉国的崛起。计算机进展国际会议论文集，2022: 1–10. https://doi.org/10.1145/3542954.3543005

Manuscript Information
Word count: 8,157 words (excluding references).

Peer-Review Record
Fast-track status: Not fast-tracked.
First-round reviews received: 3 reports.
Revision cycles completed: 3 rounds.
Final version submitted: January 5, 2026