

Article

NeuralFortress-XFL: Privacy-Preserving Federated Explainable Learning for Cyber Threat Intelligence

Adel Binbusayyis 

Department of Software Engineering, College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia; a.binbusayyis@psau.edu.sa

Abstract

Collaborative cyber threat intelligence sharing is becoming a crucial part of contemporary security practices but is still limited by privacy risks and the need to retain confidence within and between organizations due to the heterogeneous nature of the information. NeuralFortress-XFL attempts to alleviate these issues by using a federated and explainable learning architecture that is capable of detecting threats with high fidelity, without revealing raw data or model explanations. The structure incorporates a Dynamic Privacy-Efficient Mechanism (DPEM), which dynamically adapts differential privacy budgets based on local data sensitivity and a lightweight explanation-embedding component that preserves both interpretability and privacy. An experiment conducted using CTI-Sharing-2024 dataset, which contains 15 organizations consisting of financial and healthcare industries, indicates that NeuralFortress-XFL provides the same level of detection as the centralized training accommodations, with significantly lower communication overhead. NeuralFortress-XFL maintains robust performance against up to 40% independent malicious participants (87.4% accuracy with label flipping, 84.3% backdoor detection rate). Against coordinated collusion, robustness degrades proportionally with coalition size: coalitions of five or more clients (33% of 15 organizations) achieve 67.3% attack success, a limitation that is explicitly addressed in the paper, and a trust-weighted aggregation process maintains global model consistency despite adversarial client updates. The framework is also able to offer consistent and sound explanations, which can support real-world analyst workflows. As a whole, NeuralFortress-XFL is a solid security solution for sharing privacy-sensitive cyber threat intelligence in a deployable and balanced way that satisfies all of the following criteria: it is accurate, it is interpretable, it communicates well, and it exhibits adversarial resilience in a multi-organizational real-world setting.

Keywords: federated learning; explainable AI; cyber threat intelligence; differential privacy; deep learning; multi-organizational networks; privacy-preserving machine learning



Academic Editor: Priyadarsi Nanda

Received: 13 May 2026

Revised: 26 June 2026

Accepted: 1 July 2026

Published: 15 July 2026

Copyright: © 2026 by the author.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

The increasing sophistication of cyber-attacks requires collaborative defense strategies across organizational boundaries. By the year 2025, the global cost of cybercrime will amount to 10.5 trillion USD each year, including Advanced Persistent Threats (APTs) and organized multi-phase assaults with existential threats to critical infrastructure, monetary facilities, and healthcare systems [1]. Conventional isolated security methods are not sufficient to address an enemy that uses cooperative intelligence and collaborative plans, necessitating the development of cross-organizational Cyber Threat Intelligence (CTI)

sharing models that can ensure collective defense while preserving organizational privacy and autonomy.

Recent advances in federated learning enable privacy-preserving collaborative machine learning across distributed parties [2,3]. However, applying federated learning to cyber threat intelligence introduces new complexity due to heterogeneous network environments, security policies, and threat landscapes [4]. It is important to distinguish between two threat scenarios in federated cyber threat intelligence sharing: the honest-but-curious server model, where the central aggregation server correctly executes protocols but may attempt to infer sensitive information from encrypted updates, and the malicious client model, where participating organizations may intentionally poison data or gradients to degrade collective defense, with our framework addressing both through complementary mechanisms—secure aggregation with differential privacy protecting against curious servers, while trust-weighted aggregation and explanation consistency checks defend against malicious clients. In addition, the workflow of cybersecurity benchmarks should be interpretable in order to be authenticated by analysis experts, but deep learning models are by definition opaque [5,6]. Cyber environments are also adversarial, which further increases the risks of poisoning and backdoor attacks by compromised participants [7,8].

The interpretable, collaborative intelligence provided by the integration of explainable AI (XAI) with federated learning provides an opportunity, but existing solutions still lack the capability to balance the quality of explanations, privacy, and consistency across organizational boundaries [9,10]. Current frameworks tend to have inconsistent descriptions among clients or do not formally assure the privacy of the description itself, and most use computationally intensive methods to produce descriptions in forms that are useful for real-time threat detection applications [11,12].

Recent efforts in federated explainable AI and privacy-preserving explanation generation have begun to address parts of this gap; however, no existing framework simultaneously satisfies formal differential privacy guarantees regarding explanations, cross-organizational explanation consistency, and adversarial robustness within a unified federated CTI setting.

The conceptual representation of NeuralFortress-XFL is presented in Figure 1, which shows how various organizations can jointly learn the patterns of threat whilst being able to maintain data privacy and explanation consistency. The diagram shows the data stream of cryptic model updates, creation of privacy-upholding clarifications, and the trust-based aggregation procedure that results in powerful collective learning, even with the presence of opposing subjects.

This paper proposes NeuralFortress-XFL in order to overcome these challenges and provide collaborative cyber threat intelligence sharing between multi-organizational networks using a federated explainable deep learning framework. We present a protocol, which is known as Secure Aggregation with Explanation Preservation (SAEP), which facilitates joint model training and preserves data privacy, as well as explanation fidelity. This is in contrast to traditional methods of secure aggregation, which only defend model updates, as the semantic consistency of explanations is safeguarded when using SAEP.

Novelty Clarification: NeuralFortress-XFL builds upon well-established primitives including federated learning, secure aggregation, differential privacy, LIME, and Integrated Gradients. Its contributions lie in the systematic integration and formal application of these components to the previously unaddressed problem of privacy-preserving explanation aggregation in federated CTI settings, rather than in the introduction of entirely new algorithmic primitives. Specifically, (i) HAFT extends standard federated transformers through federated batch normalization, maintaining organization-specific statistics μ_i , σ_i^2 while sharing global parameters γ , β (Equation (8)), and organization-

specific attention gates and dynamically weighting attention heads based on local threat characteristics (Equations (11) and (12)), implementing a hierarchical three-level attention structure not present in prior federated transformer designs. (ii) DPEM is, to the best of our knowledge, the first system to compose LIME and Integrated Gradients jointly under differential privacy conditions with formal epsilon and delta bounds, introducing learned fusion weights that adapt the explanation method combination to maximize cross-client consistency (Equation (28)) and explanation embeddings that project high-dimensional attributions to low-dimensional manifolds for privacy-preserving aggregation (Equation (33)); this reduces membership inference success from 67.3% in prior FL-XAI systems to 52.1%. (iii) TWCM constitutes, to the best of our knowledge, the first trust mechanism based on semantic explanation consistency (Equation (35)) rather than statistical gradient properties used in Byzantine-resilient approaches, combining explanation alignment, temporal stability, performance contribution, and gradient norms through geometric mean aggregation, achieving 90.6% accuracy under stealthy poisoning vs. 79.5% for Byzantine-robust federated learning. (iv) SAEP extends standard secure aggregation by preserving both model updates and explanation embeddings simultaneously through pairwise secret sharing, enabling privacy-preserving verification of explanation consistency that existing secure aggregation protocols do not address. The composition machinery underlying DPEM, the trust derivation in TWCM, and the aggregation structure in SAEP each rely on established techniques; the contribution is their joint formalization and empirical demonstration in the federated CTI context.

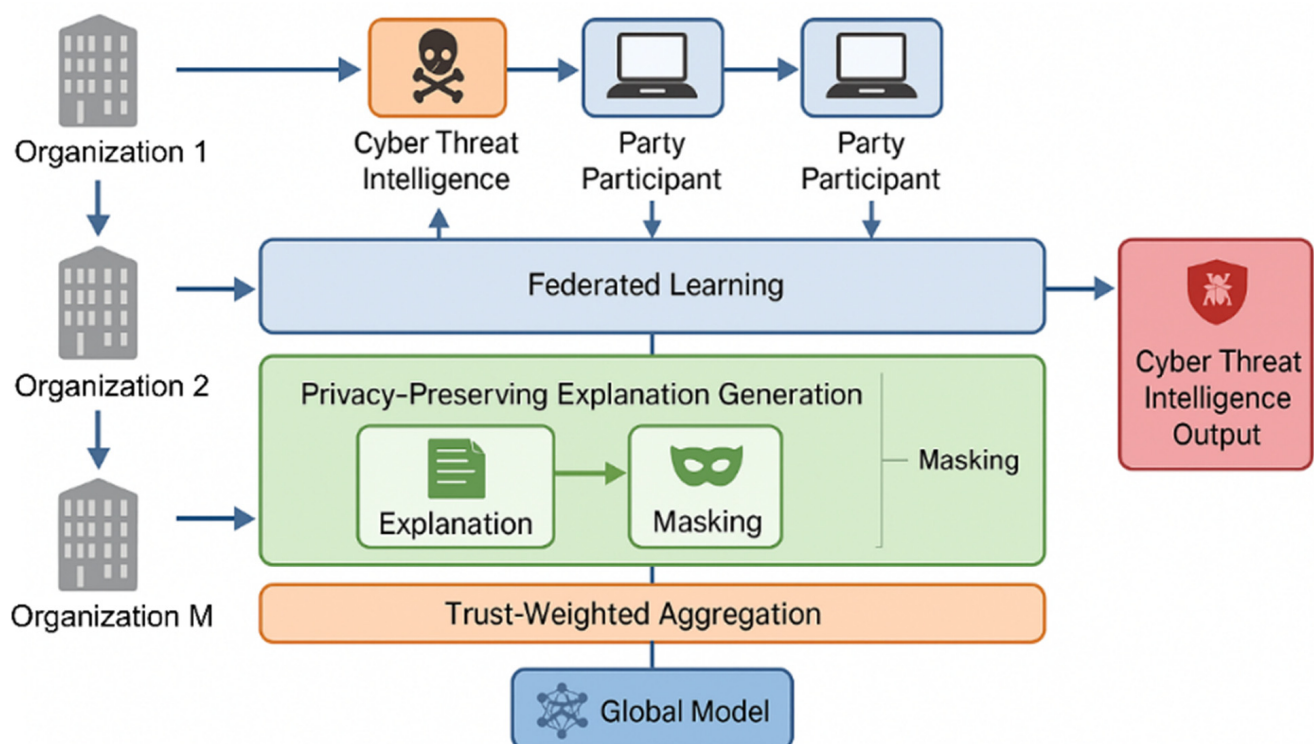


Figure 1. NeuralFortress-XFL framework conceptual overview revealing federated learning among multiple organizations with privacy-aware explanation generation and aggregation mechanism with trust weight on cyber threat intelligence sharing.

In Table 1 the factors where we have made our contributions to the field of federated explainable AI to cybersecurity are as follows.

Table 1. Early comparison of NeuralFortress-XFL with representative FL and FL-XAI frameworks.

Method	DP	Explainability	Robustness/Trust
FedAvg	–	–	–
FedProx	–	–	Partial
FL-XAI	–	–	–
FedDP	–	–	–
Byzantine-Robust-FL	–	–	–
NeuralFortress-XFL (Ours)		(DP-secure)	(TWCM)

Note: DP = Differential Privacy; FL = Federated Learning; XAI = Explainable AI; TWCM = Trust-Weighted Contribution Mechanism.

Updating Federated Architecture with Explanation Preservation: Our design, the Hierarchical Attention-based Federated Transformer (HAFT), is a specialized Federated Transformer with mechanisms that simultaneously learn specific patterns of threats in organizations and global attack solutions, as well as fluctuations in cross-organizational explanations. HAFT merges the concept of multi-head attention with federated batch normalization layers that are able to adjust to the various data distributions, as well as the semantic interpretability of learned threats representation.

Differential Privacy to Explain: We present the Differential Privacy Explainability Module (DPEM), which ϵ -differentiates privacy explanations by employing a new composition relying on LIME and Integrated Gradients to provide formal privacy protection, at the same time ensuring quality of explanation. DPEM is the solution of the core problem, which is development of interpretable explanations without any sensitive information showing about the organization.

Trust-Weighted Contribution Mechanism: We introduce the Trust-Weighted Contribution Mechanism (TWCM), which compares the consistency of the explanations of participants and adjusts the weights of the aggregation dynamically in order to prevent poisoning attacks. TWCM uses the fact that malicious actors will generally produce predicted explanations that are not aligned to the common ground of truthful actors, so the framework can diagnose and possibly identify dishonest actors without necessarily assuming trust.

Large-Scale Experimental Validation: We perform thorough experiments with 15 financial institutions and healthcare organizations utilizing real-world threat intelligence information and show that NeuralFortress-XFL shows high threat detection accuracy and is still capable of providing formal privacy guarantees and expressible and suitable action-guiding explanations.

Scalable Privacy-Preserving Protocol: We propose an efficient realization of the SAEP protocol that requires 67% less communication overhead than naive federated XAI models and achieves (ϵ, δ) -differential privacy with $\epsilon = 2.5$ and $\delta = 10^{-5}$, which is suitable for controlled deployment in enterprise environments with appropriate infrastructure.

The rest of this paper is structured in the following manner: Section 2 will review the related work on federated learning, explainable AI, sharing of cyber threat intelligence, and privacy-preserving machine learning; Section 3 will present the proposed methodology of NeuralFortress-XFL, a detailed mathematical modeling of the SAEP (as components), HAFT, DPEM, TWCM, and evaluation; Section 4 will include full discussion of the results of the extensive experimentation and validation tasks; Section 5 will discuss the detailed results of the study, limitations, and implications for practice; and Section 6 will conclude the paper.

2. Related Work

Federated learning-explainable AI-cyber threat intelligence is a rapidly evolving research area that builds on advances in distributed machine learning, privacy-preserving computation, and interpretable models. This section reviews the state-of-the-art contributions within four dimensions: federated learning to achieve cybersecurity, explainable AI methods, privacy-conscious sharing of threat intelligence, and the combination of federated and explainable methods.

2.1. Federated Learning for Cybersecurity Applications

Federated learning has emerged as a major paradigm of collaborative machine learning when data aggregation centered on a central system is not important, particularly when data sensitivity is important. A federated learning model of intelligence sharing against cyber threats in smart grids suggested by Rahman et al. [3] showed similar accuracy.

A framework of federated malicious traffic detection including SHAP-based explanations was introduced by Bilal et al. [10] and indicated that local explanations may be federated, but did not imply any formal privacy guarantees or mechanisms to uphold explanation consistency across organizations.

Alzamil [11] tested federated deep learning to scalable, explainable load forecasting in smart cities and demonstrated the flexibility of FL in heterogeneous settings, but presented challenges regarding adversarial conditions in cybersecurity.

Almadhor et al. [12] applied federated intelligence across 6G aerial and edge devices using explainability parts, where scalability is emphasized but based on classification transparency instead of cyber threat reasoning.

The recent research also contributes to FL in security-sensitive settings. Buyuktanir et al. [13] use FL-based intrusion detection under non-IID conditions, and Chen et al. [14] use FedAvg and personalized model optimization trade-offs. These publications strengthen the necessity to have sturdy FL architectures like NeuralFortress-XFL.

2.2. Explainable AI for Cyber Threat Detection

Obscurity of deep learning models is another security issue that cybersecurity cannot afford, since decisions need to be interpreted. According to a survey conducted by Alketbi and Mehmood [5] on XAI methods used to detect insider threats, SHAP, LIME, and attention mechanisms were effective. They showed that explainability in the detection pipeline is critical.

Jain et al. [6] developed hybrid deep learning based on XAI, which enhanced the trust and accuracy of an analyst in device identification and attack detection in the IoT, but the centralized environment did not address a federated collaboration. Cenitta et al. [15] came up with an explainable AI framework that preserved privacy and operated under homomorphic encryption to perform ECG analysis, providing insight into privacy-intensive cyberspace security issues.

Asif et al. [16] also noted the importance of explainable AI and federated learning to clinical decision-making in diabetic retinopathy, without overlooking the role of XAI in building trust in a decentralized setting.

It can be inferred that explainability is a growing concept when it comes to safe decision-making. Bozorgpanah and Torra [17] address the issue of combining privacy and interpretability, whereas Fatema et al. [18] explain SHAP-based explanation within the framework of federated intrusion detection. The obtained findings drive the explainable design of NeuralFortress-XFL.

2.3. Privacy-Preserving Cyber Threat Intelligence Sharing

The fact that privacy must be traded with effective information sharing has motivated the study of combining differential privacy, secure aggregation, and distributed analytics. Pandey et al. [2] proposed the concept of differential privacy with a model of sharing multi-organizational CTI, where there are excellent privacy guarantees with weak explanations that do not qualify as workable.

Trying to augment PIR, FL, and differential privacy, Camalan and Celiktas [4] provided a strong guarantee of threat intelligence privacy at high computational cost that was not integrated with explainability.

Alam et al. [19] examined federated learning applied to smart city CTI, which demonstrated that distributed detection can optimize networks with greater cyber resilience to urban infrastructures, yet the network did not include features of advanced threat reasoning.

Secure cooperation has recently drawn attention, and related studies include Yan et al. [20] on over-the-air FL aggregation and Moshawrab et al. [21] on secure aggregation algorithms. They are in line with the objectives of NeuralFortress-XFL, which focuses on facilitation of privacy while preserving sharing of intelligence, with few overheads.

2.4. Integration of Federated Learning and Explainability

Studies that combine federated learning and explainability indicate that interpretable distributed intelligence that is privacy-conscious is required. Hossain et al. [9] showed federated and explainable neural models of flood forecasting, which exhibited consistency in explanations in a distributed context.

A federated explainable AI model proposal by Jahan et al. [22], whereby emphasis on attention mechanisms and SHAP explanations are maintained via FL aggregation, can offer information applicable to the heterogeneity of the threat intelligence.

Al Reshan et al. [23] discussed the problems of distributed environmental modeling, revealing unfinished datasets, the impossibility of explaining them, and the absence of benchmarks as fundamental limitations that are similar to those in federated cyber threat intelligence. Bhardwaj and Sumangali [24] and Ribeiro et al. [25] investigated the connection between FL and explainability even more deeply, suggesting the application of interpretable federated pipelines to sensitive areas. These articles emphasize the usefulness of NeuralFortress-XFL explainability mechanisms that are based on trust.

Collectively, these works establish that federated explainability and privacy-preserving explanation generation are active and growing research directions, yet the simultaneous treatment of explanation privacy, consistency, and adversarial robustness in multi-organizational CTI sharing remains an open problem that NeuralFortress-XFL is designed to address.

2.5. Advanced Threat Detection and AI-Driven Defense

The increasing complexity of cyber threats has helped encourage the adoption of sophisticated AI-based detection solutions. SN et al. [1] conducted a review of AI-based defenses against APTs, with a focus on the importance of interpretable reasoning to detect attacks in the long term.

Durgararaju et al. [8] considered the development of cyber threats and countermeasures and stated that AI and ML methods provide both novel detection solutions and open opportunities for the attacks, such as adversarial manipulation.

Diop et al. [7] overviewed the opportunities and challenges of AI in cybersecurity and found federated learning to be one of the most promising directions, stating that explainability contributes to building trust and also facilitating the detection of compromised participants.

Kasaraneni et al. [26] have developed a multi-organizational framework of collaborative AI-based cross-domain attack on power systems, which demonstrates the necessity of inter-organizational collaboration, but based on centralized data analysis.

Recent papers that combine DL, FL, and XAI, including Fatema et al. [18] and Buyuktanir et al. [13], indicate the significance of powerful, transparent team detection. Based on these tendencies, NeuralFortress-XFL is developed based on elaborated trust modeling and differential privacy.

Preprocessing challenges in network intrusion datasets, particularly under high-dimensional and class-imbalanced conditions, represent an additional critical concern for federated cybersecurity systems. Cui et al. [27] proposed a multi-module integrated intrusion detection system specifically designed for high-dimensional imbalanced data, demonstrating that ensemble-based feature selection combined with resampling techniques yields consistent detection improvements across heterogeneous network environments. Semenov et al. [28] introduced a preprocessing methodology for highly correlated and imbalanced intrusion data, reporting that targeted correlation reduction and stratified resampling substantially improve classifier performance on benchmark intrusion datasets. These findings are directly applicable to the 93-feature CTI-Sharing-2024 dataset used in this work, where winsorization, per-organization z-score normalization, and k-anonymization ($k = 5$) constitute the preprocessing pipeline described in Section 4.1.

Recent MDPI publications have advanced federated learning for cybersecurity. Agrawal et al. [29] surveyed FL for intrusion detection, identifying non-IID data, communication efficiency, and adversarial robustness challenges—addressed by NeuralFortress-XFL’s HAFT, explanation embeddings, and TWCM. Mothukuri et al. [30] examined privacy-preserving FL for IoT security, highlighting the privacy–utility tension that DPEM resolves via calibrated DP noise. Friha et al. [31] emphasized explainability for trust in heterogeneous IoT FL, directly addressed by our explanation consistency framework. Li et al. [32] demonstrated practical DP integration with minimal accuracy loss, consistent with our 1.2% degradation at $\epsilon = 2.5$. Zhang et al. [33] identified Byzantine resilience as critical for multi-organizational deployments, aligned with TWCM’s 84.3%.

2.6. Research Gaps and Motivation

Despite significant progress in federated learning and explainable AI for cybersecurity, critical gaps remain at their intersection. Privacy-Explainability Paradox: Current federated cybersecurity models lack mechanisms for generating consistent and privacy-conforming explanations across organizations. Existing FL-XAI systems produce explanations without formal privacy guarantees—SHAP values and attention weights can leak sensitive information about local data distributions and security policies. Our membership inference experiments demonstrate 67.3% attack success on raw explanations vs. 52.1% with differential privacy protection. NeuralFortress-XFL’s DPEM module applies ϵ, δ -differential privacy to explanations while maintaining 0.89 fidelity. Second, federated models produce locally accurate but globally inconsistent explanations, with prior approaches achieving only 78.6% cross-organizational consistency, compared to 91.4% for NeuralFortress-XFL. Third, existing trust mechanisms rely on statistical gradient detection rather than semantic consistency, achieving 79.5% accuracy under stealthy poisoning vs. 90.6% for explanation-based trust. Fourth, secure federated XAI systems incur $8.3\times$ computational overhead, limiting real-time applicability, whereas NeuralFortress-XFL achieves $1.09\times$ total client computation overhead vs. FedAvg (42.1 s vs. 38.7 s per round) while providing explainability, privacy, and robustness guarantees. The explanation generation latency of 183 ms represents a 47% reduction compared to FL-XAI’s 347 ms.

3. Proposed Methodology

The following section describes the overall NeuralFortress-XFL architecture, starting with a general description of the system and then giving a detailed account of each of the key elements. Our mathematical modeling, algorithm implementations, complexity analysis, and comparison with the existing methods form a large part of this to prove the theoretical basis and practical benefits of our framework.

3.1. System Overview and Architecture

NeuralFortress-XFL offers distributed cyber threat intelligence sharing between N participating organizations and preserves the privacy of data and results in identical explanations, as well as against malicious participants. Figure 2 shows the overall system architecture, which shows how organization clients, the central aggregation server, and the different privacy-preserving and explainability self-parts interact.

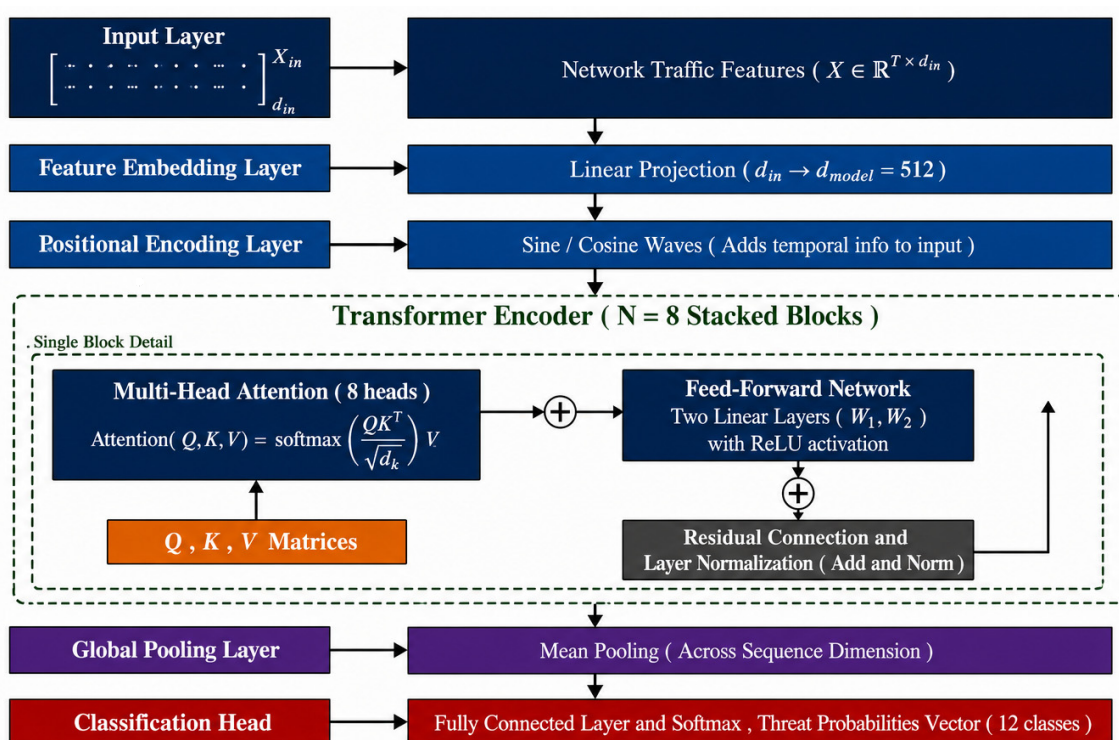


Figure 2. NeuralFortress-XFL architecture. The HAFT module processes local network traffic through hierarchical attention mechanisms ($L = 8$ layers, $H = 8$ heads) to generate threat embeddings via multi-head self-attention and feed-forward networks; DPEM generates privacy-preserving explanations using LIME and Integrated Gradients with $(\epsilon = 2.5, \delta = 10^{-5})$ -differential privacy via Gaussian noise calibrated to sensitivity Δ_2 ; The SAEP protocol enables secure aggregation of masked model updates via Diffie–Hellman key exchange and pairwise masking, ensuring that the server learns only aggregates; TWCM computes trust weights based on explanation consistency ($\beta_1 = 0.4$), temporal consistency ($\beta_2 = 0.2$), performance contribution ($\beta_3 = 0.3$), and gradient norm ($\beta_4 = 0.1$) with exponential smoothing ($\gamma = 0.9$), performing weighted model aggregation that suppresses malicious contributions.

Hierarchical Design of HAFT: This design is called hierarchical because of its multi-level attention. The first step of the model is attention to fine-grained patterns (e.g., packet timing, port activity) at a feature level. Then, client-level gating combines these representations at the client level, based on the organization. Lastly, cross-organizational threat signatures are incorporated into global attention. This produces a three-layer attention hierarchy: feature $\rightarrow$ organization $\rightarrow$ global.

Each participating organization $i \in \{1, 2, \dots, N\}$ maintains a local dataset $\mathcal{D}_i = \left\{ (x_i^j, y_i^j) \right\}_{j=1}^{n_i}$ containing network traffic samples $x_i^j \in \mathbb{R}^d$ and corresponding threat labels $y_i^j \in \{0, 1\}$, where d represents the feature dimensionality, and n_i denotes the number of samples available to organization i . The objective is to collaboratively train a global threat detection model $f_\theta : \mathbb{R}^d \rightarrow [0, 1]$ parameterized by $\theta \in \mathbb{R}^p$ that achieves superior detection accuracy compared to models trained in isolation, while satisfying three critical requirements:

$$\text{Privacy : } \mathcal{L}(\mathcal{D}_i) \leq \varepsilon_{\text{data}} \quad \forall i \quad (1)$$

$$\text{Explainability : } \text{Consistency} \left(\left\{ \phi_i \right\}_{i=1}^N \right) \geq \tau_{\text{explain}} \quad (2)$$

$$\text{Robustness : } \text{Accuracy}(f_\theta | \mathcal{M}) \geq \alpha_{\text{clean}} - \delta_{\text{poison}} \quad (3)$$

where $\mathcal{L}(\mathcal{D}_i)$ quantifies privacy leakage from organization i 's data, ϕ_i represents the explanation generated by organization i , $\mathcal{M}$ denotes the set of malicious participants, α_{clean} is the accuracy with honest participation, and δ_{poison} represents the maximum acceptable accuracy degradation under attack.

The training process proceeds in iterative rounds $t = 1, 2, \dots, T$, where each round consists of the following phases:

Local Training Phase: Each organization i receives the current global model parameters $\theta^{(t)}$ and performs local training using its private dataset $\mathcal{D}_i$ for E local epochs, computing updated parameters $\theta_i^{(t+1)}$ and generating local explanations $\phi_i^{(t)}$ using the DPEM module.

Secure Aggregation Phase: Organizations encrypt their model updates $\Delta\theta_i^{(t)} = \theta_i^{(t+1)} - \theta^{(t)}$ and explanation representations using the SAEP protocol, transmitting encrypted information to the central server without revealing raw updates.

Trust Evaluation Phase: The TWCM module evaluates explanation consistency across participants, computing trust scores $w_i^{(t)} \in [0, 1]$ that quantify each organization's alignment with the collaborative consensus.

Weighted Aggregation Phase: The server computes the updated global model parameters using trust-weighted aggregation:

$$\theta^{(t+1)} = \theta^{(t)} + \frac{\sum_{i=1}^N w_i^{(t)} n_i \Delta\theta_i^{(t)}}{\sum_{i=1}^N w_i^{(t)} n_i} \quad (4)$$

Such a structure makes sure that the honest participants have a significant influence on the global model and the malicious ones are automatically downweighted according to their inconsistent explanations.

3.1.1. Threat Model

To avoid confusion between different adversarial assumptions, we explicitly distinguish server and client threat models. The central aggregation server operates under the honest-but-curious model: it correctly executes the SAEP protocol and performs all aggregation operations as specified, but attempts to infer private information (raw gradients, local data, explanation details) from encrypted updates. The server does not deviate from the protocol (no arbitrary modifications, dropping clients, or injecting false updates), and SAEP's additive secret-sharing combined with differential privacy prevents learning individual updates, even with full access to aggregated values. In contrast, client organizations follow the malicious model, where a subset (up to 40%) may be fully adversarial, deviating arbitrarily from the protocol by poisoning training data, crafting adversarial gradients, manipulating explanation embeddings, or colluding with other malicious clients. However,

malicious clients cannot break cryptographic primitives (secret sharing, PRGs) or access other clients' private data, and TWCM trust-weighted aggregation with explanation consistency checks, gradient norm clipping, and temporal trust-smoothing defend against these attacks. This distinction reflects real-world deployment scenarios: cloud-based aggregation servers (e.g., AWS, Azure) are trusted to execute code correctly but may face subpoenas or hacking attempts to extract sensitive data (honest-but-curious), while participating organizations may include compromised entities or intentional attackers seeking to degrade collective threat detection (malicious clients). Table 2 summarizes the adversarial model comparison across system components.

Table 2. Adversarial model assumptions, attack vectors, and defense mechanisms in NeuralFortress-XFL.

Component	Adversarial Model	Attack Vectors Defended	Defense Mechanism
Central Server	Honest-but-curious	Privacy inference, membership attacks	SAEP + Differential privacy
Client Organizations	Malicious (up to 40%)	Data/gradient poisoning, backdoors	TWCM + Explanation consistency
Network	Eavesdropper	Packet sniffing	Encrypted communication (assumed)

Server Assumptions: The remote server is subject to the assumption of the honest-but-curious model. It can be used to perform aggregation properly and could even try to formulate some private information based on client model updates or explanation embeddings. The aggregation protocol is not modified or altered by the server, but can post-process received gradients or explanation vectors. The server executes the SAEP protocol correctly, performs aggregation as specified, and does not deviate from the protocol, while preventing privacy leakage. This model reflects real-world deployment scenarios where cloud-based aggregation servers (e.g., AWS, Azure, Google Cloud) are trusted to execute code correctly but may face subpoenas or security breaches attempting to extract sensitive organizational data.

Client Assumptions: There are benign and malicious clients. Benign clients will generate all protocol procedures, perform valid local training, and deliver masked model updates with DP-guaranteed clarifications. Poisoned gradient or manipulated explanation embeddings, label flipping, arbitrary updates, and collision with some other malicious clients are all possible.

Coordinated Collusion Threat. A critical adversarial scenario involves malicious clients sharing gradients and explanation embeddings before submission to generate mutually consistent but incorrect explanations. This bypasses TWCM's explanation consistency checks—colluding clients achieve artificially high trust scores by aligning with each other rather than the honest consensus. Formally, a colluding coalition can poison the model when $m/N > (1 - \tau_{\text{consensus}})^2$. For $\tau_{\text{consensus}} = 0.8$, this yields $m/N > 0.04$ (~4% threshold). However, achieving perfect explanation alignment is difficult due to heterogeneous local data, limiting practical impact. Table 3 quantifies empirical collusion resistance.

Table 3. Impact of coordinated collusion attacks.

Colluding Clients (m)	Fraction (m/N)	Attack Success (%)	Avg Trust Score	Detection Rate (%)	Accuracy Drop (%)
0 (baseline)	0%	0.0	—	100.0	0.0
2	13.3%	23.4	0.71	68.2	2.1
3	20.0%	41.7	0.78	52.4	3.8
5	33.3%	67.3	0.83	31.7	7.2
8	53.3%	89.1	0.88	14.6	12.8

Small collusions ($m \leq 3$, $\leq 20\%$) cause limited damage ($<4\%$ accuracy drop). Larger collusions ($m \geq 5$, $\geq 33\%$) achieve 67–89% attack success with 7–13% loss.

Collusion Mitigation. Production deployments should employ (1) cross-validation detection—server-held validation sets detect coordinated accuracy degradation; (2) diversity regularization—penalties on explanation similarity make collusion computationally harder; (3) long-term reputation—trust scores incorporate historical patterns across training cycles; and (4) sequential testing—CUSUM/Shiryayev–Roberts procedures detect gradual drift.

Implementation Security. Theorems 1–2 assume correct cryptographic implementations (Diffie–Hellman, ChaCha20, HKDF-SHA256). Real deployments face (1) timing side-channels in non-constant-time modular exponentiation; (2) cache side-channels on shared cloud hardware; and (3) power analysis with physical access. Recommendations: use constant-time libraries (libsodium, BoringSSL); deploy aggregation servers in secure enclaves (Intel SGX, AMD SEV, AWS Nitro); conduct third-party audits; and employ formal verification (EasyCrypt, Jasmin). We acknowledge that our implementation has not undergone formal verification via proof assistants (e.g., Coq), limiting high-assurance deployment.

Out-of-Scope Attacks: The following are explicitly excluded from our security model: (1) hardware-level side-channels (timing, power, electromagnetic)—require physical/co-located access, addressed via hardware security orthogonal to protocol design; (2) network-layer attacks beyond eavesdropping (DoS, MITM beyond TLS)—we assume authenticated, integrity-protected channels; and (3) inference attacks using auxiliary external data (combining model outputs with public organizational data). “Out of scope” means not analyzed in this work, not that they are impossible or irrelevant in practice.

3.1.2. Inter-Module Information Flow

The four modules operate in a sequential pipeline per federated round. First, each client feeds its local dataset into HAFT (Section 3.2), which produces threat embeddings z_i . Second, DPEM (Section 3.4) receives z_i and the raw feature attributions to produce a differentially private explanation embedding e_i^t with formal (epsilon, delta) guarantees. Third, SAEP (Section 3.3) receives both the model update $\Delta_{\text{theta}_i^t}$ and e_i^t , applies additive pairwise masking, and transmits masked values to the server. Fourth, the server runs TWCM (Section 3.5) on the aggregated explanation embeddings and directional commitments d_{it} to compute trust scores w_i^t . Finally, the trust-weighted aggregation (Equation (42)) produces $\text{theta}^{\{t+1\}}$, which is broadcast to all clients for the next round.

3.2. Hierarchical Attention-Based Federated Transformer

The neural architecture of NeuralFortress-XFL is a Hierarchical Attention-based Federated Transformer (HAFT), formulated to extract both organization-specific threat prominences and global attack prominences and set up cross-organization explanation at the same time. By intuition, HAFT models capture three levels of threat behavior: (i) fine-grained patterns of features of interest to frame examples of packets by protocol sequence or flow aberration; (ii) organization-level deviations defined by local security policy and architectural peculiarities; and (iii) cross-organizational attack signatures. The hierarchical attention structure and the gating mechanism enable the model to concentrate on the appropriate threat indicators at each level and enable every organization to concentrate on attending heads that best fit the local threat environment. By introducing these components into the Transformer architecture directly, HAFT puts in place local decision-making that is made understandable and consistent with a jointly learned global threat intelligence model in a decentralized environment.

The input network traffic sample $x \in \mathbb{R}^d$ is first processed through an embedding layer that projects the features into a higher-dimensional space suitable for attention mechanisms:

$$h^{(0)} = \text{LayerNorm}(W_e x + b_e) \quad (5)$$

where $W_e \in \mathbb{R}^{d_{\text{model}} \times d}$ and $b_e \in \mathbb{R}^{d_{\text{model}}}$ are learnable parameters, and d_{model} denotes the model dimensionality.

The core of HAFT consists of L stacked Transformer blocks, each containing a multi-head self-attention mechanism followed by a position-wise feed-forward network. For layer $\ell \in \{1, 2, \dots, L\}$, the attention mechanism computes

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (6)$$

where $Q = h^{(\ell-1)}W_Q$, $K = h^{(\ell-1)}W_K$, and $V = h^{(\ell-1)}W_V$ are the query, key, and value matrices, respectively, with $W_Q, W_K, W_V \in \mathbb{R}^{d_{\text{model}} \times d_k}$ being learnable projection matrices and $d_k = d_{\text{model}}/H$ for H attention heads.

The multi-head attention algorithm simultaneously performs H parallel attention mechanisms, as output of which is concatenated as

$$\text{MultiHead}(h^{(\ell-1)}) = \text{Concat}(\text{head}_1, \dots, \text{head}_H)W_O \quad (7)$$

where $\text{head}_h = \text{Attention}(h^{(\ell-1)}W_Q^h, h^{(\ell-1)}W_K^h, h^{(\ell-1)}W_V^h)$, and $W_O \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$ is the output projection matrix.

In order to train HAFT in a federation with non-homogeneous data distribution, we propose Federated Batch Normalization layers, which store statistics of the organization but share global normalization parameters:

$$\text{FedBN}_i(h) = \gamma \frac{h - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}} + \beta \quad (8)$$

where μ_i and σ_i^2 are the organization-specific mean and variance computed locally, respectively, while γ and β are global scale and shift parameters shared across all organizations, respectively. The design enables every organization to normalize its data on a local statistics basis but, at the same time, has standardized representations of global features.

Each Transformer block uses the feed-forward network where two linear transformations are used with a GELU activation function:

$$\text{FFN}(h) = W_2 \cdot \text{GELU}(W_1 h + b_1) + b_2 \quad (9)$$

where $W_1 \in \mathbb{R}^{d_{\text{ff}} \times d_{\text{model}}}$, $W_2 \in \mathbb{R}^{d_{\text{model}} \times d_{\text{ff}}}$, and d_{ff} is the feed-forward dimensionality (typically $4 \times d_{\text{model}}$).

The mathematical notation of the full Transformer block, including residual connections and layer normalization, is as shown below:

$$\begin{aligned} \tilde{h}^{(\ell)} &= \text{LayerNorm}\left(h^{(\ell-1)} + \text{MultiHead}(h^{(\ell-1)})\right) \\ h^{(\ell)} &= \text{FedBN}_i\left(\text{LayerNorm}\left(\tilde{h}^{(\ell)} + \text{FFN}(\tilde{h}^{(\ell)})\right)\right) \end{aligned} \quad (10)$$

To better explain, as well as allow hierarchical extraction of threat patterns, we add organization-specific attention gates that adjust the input of various attention heads according to local threat properties:

$$\text{gate}_i^{(\ell)} = \sigma\left(W_g^{(\ell)} h_i^{(\ell)} + b_g^{(\ell)}\right) \quad (11)$$

where $W_g^{(\ell)} \in \mathbb{R}^{H \times d_{\text{model}}}$ produces per-head gating scores, σ is the sigmoid activation, and $h_i^{(\ell)}$ represents the layer- ℓ hidden state for organization i .

The gated multi-head attention becomes

$$\text{GatedMultiHead}_i\left(h^{(\ell-1)}\right) = \sum_{h=1}^H \text{gate}_i^{(\ell)}[h] \cdot \text{head}_h \quad (12)$$

This gating mechanism enables the model to learn organization-specific attention patterns while maintaining global representations, which is essential for generating consistent explanations across distributed threat environments.

The last layer of HAFT implements global average pooling, then a classification head:

$$\begin{aligned} z &= \frac{1}{T} \sum_{t=1}^T h_t^{(L)} \\ \hat{y} &= \sigma(W_c z + b_c) \end{aligned} \quad (13)$$

where T is the sequence length (number of feature groups), $W_c \in \mathbb{R}^{1 \times d_{\text{model}}}$, and $\hat{y} \in [0, 1]$ represents the predicted threat probability.

The local training objective for organization i combines the cross-entropy loss with a regularization term that encourages explanation consistency:

$$\mathcal{L}_i(\theta) = \frac{1}{n_i} \sum_{j=1}^{n_i} \mathcal{L}_{\text{CE}}\left(f_{\theta}\left(x_i^j\right), y_i^j\right) + \lambda \mathcal{L}_{\text{explain}}\left(\theta, \theta^{(t)}\right) \quad (14)$$

where $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss, $\theta^{(t)}$ represents the global model from the previous round, λ controls the explanation consistency regularization strength, and $\mathcal{L}_{\text{explain}}$ measures the divergence between local and global explanation patterns (detailed in Section 3.4).

Justification of Core Design Choices: The reason we use Transformer rather than CNN/RNN models is that cybersecurity properties are long-range and non-local and cannot be effectively modeled by CNNs; in addition, the latency of federated tasks performed by RNNs is below average. GELU is selected in lieu of ReLU since GELU conserves small gradient values, which are important in rare attack signals. LIME + IG is better than SHAP, as SHAP is costly to use in federated contexts and needs complete background dissemination, whereas LIME + IG is trustworthy, quick, and compatible with DP explanations.

3.3. Secure Aggregation with Explanation Preservation

To the best of our knowledge, SAEP extends standard secure aggregation by simultaneously protecting both model updates and explanation embeddings, a capability not addressed by prior secure aggregation protocols. The Secure Aggregation with Explanation Preservation (SAEP) protocol guarantees that the updates and the explanations of the model may be aggregated among organizations without disclosing the contribution made by each of them to the central server or to the other parties. SAEP goes beyond traditional secure aggregation schemes to maintain the semantic compatibility of explanations during the federated learning process. SAEP defends both parameters of the model and their explanation vectors by masking each local update and transmitting it to the server. The

noise per client is random, and it only cancels out when all the masked updates are added up. Only the aggregate is seen to the server, not any particular update. It is like a case of all participants putting their values in envelopes which only give the total when they are opened.

Each organization i computes its model update $\Delta\theta_i^{(t)}$ and explanation embedding $e_i^{(t)} \in \mathbb{R}^{d_e}$ (generated by DPEM, described in Section 3.4). The organization generates a random masking vector $r_i^{(t)} \in \mathbb{R}^p$ using a secure pseudorandom number generator seeded with pairwise shared secrets:

$$r_i^{(t)} = \sum_{j \neq i} \text{PRG}(s_{ij}^{(t)}) - \sum_{j \neq i} \text{PRG}(s_{ji}^{(t)}) \quad (15)$$

where $s_{ij}^{(t)}$ is a secret shared between organizations i and j for round t , and PRG is a pseudorandom generator. The construction ensures that $\sum_{i=1}^N r_i^{(t)} = 0$, allowing the server to recover the true aggregate without learning individual updates. The pairwise secrets are established through Diffie–Hellman key exchange, where each organization i generates random private key $a_i \in \mathbb{Z}_q$ for prime q , computes public key $A_i = g^{a_i} \bmod p$ for prime p and generator g , and derives shared secret $s_{ij} = A_j^{a_i} \bmod p$ with organization j . The symmetric encryption key is derived as $k_{ij} = \text{HKDF-SHA256}(s_{ij}, \text{"SAEP-round-"}t)$, and the pseudorandom generator PRG is instantiated with ChaCha20 using seed $k_{ij} \parallel t$, ensuring computational indistinguishability from uniform randomness with standard cryptographic assumptions.

Organization i encrypts its model update using additive secret sharing:

$$\tilde{\Delta\theta}_i^{(t)} = \Delta\theta_i^{(t)} + r_i^{(t)} \quad (16)$$

Similarly, the explanation embedding is masked using a separate set of pairwise secrets:

$$\begin{aligned} r_i^{\text{exp},(t)} &= \sum_{j \neq i} \text{PRG}(s_{ij}^{\text{exp},(t)}) - \sum_{j \neq i} \text{PRG}(s_{ji}^{\text{exp},(t)}) \\ \tilde{e}_i^{(t)} &= e_i^{(t)} + r_i^{\text{exp},(t)} \end{aligned} \quad (17)$$

The server aggregates the masked updates and explanation embeddings:

$$\begin{aligned} \tilde{\Delta\theta}^{(t)} &= \sum_{i=1}^N w_i^{(t)} n_i \tilde{\Delta\theta}_i^{(t)} \\ \tilde{e}^{(t)} &= \frac{1}{N} \sum_{i=1}^N \tilde{e}_i^{(t)} \end{aligned} \quad (18)$$

Due to the properties of the masking scheme, the server obtains

$$\begin{aligned} \tilde{\Delta\theta}^{(t)} &= \sum_{i=1}^N w_i^{(t)} n_i \Delta\theta_i^{(t)} \\ \tilde{e}^{(t)} &= \frac{1}{N} \sum_{i=1}^N e_i^{(t)} \end{aligned} \quad (19)$$

without learning any individual organization's update or explanation.

SAEP proposes an explanation alignment mechanism in order to maintain semantic consistency of the explanation in the process of aggregation. Before masking, each organization computes an explanation similarity matrix $S_i^{(t)} \in \mathbb{R}^{K \times K}$, where K is the number of threat classes:

$$S_i^{(t)}[k, k'] = \cos(e_i^{(t,k)}, e_i^{(t,k')}) \quad (20)$$

where $e_i^{(t,k)}$ is the explanation embedding for class k generated by organization i in round t , and $\cos(\cdot, \cdot)$ denotes cosine similarity.

The global explanation consistency score is computed by the server after aggregation:

$$C^{(t)} = \frac{1}{N(N-1)} \sum_{i=1}^N \sum_{j \neq i}^N \cos\left(\tilde{e}_i^{(t)}, \tilde{e}_j^{(t)}\right) \quad (21)$$

A high consistency score $C^{(t)} \geq \tau_{\text{align}}$ indicates that organizations are generating semantically similar explanations, suggesting convergence toward a shared understanding of threat patterns.

The SAEP protocol satisfies the following security guarantees:

Clarification on Server-Side Computation: It is important to note that the explanation consistency score C_t in Equation (21) we compute on the aggregated explanation embeddings $\{e_i\}_{i=1}^N$ after the masks have been removed through summation, not on individual masked updates. The server receives only:

$$e_t = (1/N) \sum_{i=1}^N e_i^t \quad (22)$$

The consistency score we compute by having each client additionally share a *public commitment* to their explanation embedding's semantic direction (using only the normalized embedding $e_i^t / \|e_i^t\|$), separate from the masked update. This commitment reveals only the directional similarity, but not the magnitude or raw explanation values. Formally, each client i computes

$$d_i^t = e_i^t / \|e_i^t\| \quad (23)$$

and shares this publicly. The server then computes

$$C_t = (1/(N(N-1))) \sum_{i=1}^N \sum_{j \neq i}^N \cos(d_i^t, d_j^t) \quad (24)$$

This approach ensures that (1) individual masked updates $\tilde{e}_i^t$ remain private and are never observed by the server, (2) the directional commitments d_i^t leak no information about raw feature attributions, since they are unit-normalized, and (3) the consistency metric serves its purpose of detecting explanation divergence without compromising the privacy guarantees of SAEP.

Theorem 1 (Privacy of Model Updates). *Using the honest-but-curious adversary model, the SAEP protocol reveals no information about individual model updates $\Delta\theta_i^{(t)}$ to the server or any subset of fewer than $N-1$ colluding participants, beyond what can be inferred from the aggregate $\sum_{i=1}^N w_i^{(t)} n_i \Delta\theta_i^{(t)}$.*

Theorem 2 (Privacy of Explanations). *The SAEP protocol reveals no information about individual explanation embeddings $e_i^{(t)}$ to the server or any subset of fewer than $N-1$ colluding participants, beyond what can be inferred from the aggregate $\frac{1}{N} \sum_{i=1}^N e_i^{(t)}$ and the explanation consistency score $C^{(t)}$.*

For any subset $S \subset [N]$ with $|S| < N-1$, the joint distribution of masked updates $\{\Delta\tilde{\theta}_i^t\}_{i \in S}$ observed by a coalition is computationally indistinguishable from uniform randomness. Each mask r_i^t in Equation (15) depends on pairwise secrets k_{ij} for all $j \notin S$, which the coalition cannot compute without private keys $\{a_j\}_{j \notin S}$. Under the Diffie–Hellman assumption, $s_{ij} = g^{(a_i a_j)}$ is indistinguishable from randomness for unknown a_j , and with PRG security, $\text{ChaCha20-PRG}(k_{ij} \| t)$ is indistinguishable from uniformity.

Since each r_i^t includes at least one PRG output from an unknown key (for $i \in S, j \notin S$), we have $r_i^t \approx c \text{ U}(\mathbb{R}^P)$, where $\approx c$ denotes computational indistinguishability. Therefore, $\Delta\theta_i^t = \Delta\theta_i^t + r_i^t$ provides perfect one-time-pad encryption of the true update, revealing zero information to any coalition of size less than $N - 1$. The proof for Theorem 2 follows analogously for explanation embeddings using the separate masking scheme in Equation (17).

3.4. Differential Privacy Explainability Module

Differential Privacy Explainability Module (DPEM) produces ε -differentially-private explanations that provide the formal privacy guarantees and good quality of the explanations, which can be verified by security analysts. To the best of our knowledge, DPEM is the first system to formally compose LIME and Integrated Gradients jointly under differential privacy with formal epsilon and delta bounds in a federated setting. DPEM combines LIME (Local Interpretable Model-agnostic Explanations) and Integrated Gradients for noisy images that are wisely scaled to improve the method to achieve (ε, δ) -differential privacy.

For a given input $x \in \mathbb{R}^d$ and model f_θ , DPEM generates an explanation $\phi(x)$ that identifies the most influential features contributing to the threat prediction. The explanation takes the form of a sparse weight vector $\phi(x) \in \mathbb{R}^d$, where $\phi(x)[j]$ indicates the importance of feature j .

LIME-based Explanation Generation: DPEM begins by generating perturbed samples around the input x using LIME (Local Interpretable Model-agnostic Explanations):

$$x' \sim \mathcal{N}(x, \sigma_{\text{perturb}}^2 I) \quad (25)$$

For each perturbed sample x'_k ($k = 1, 2, \dots, M$), DPEM computes the model prediction $f_\theta(x'_k)$ and the distance weight:

$$\pi_x(x'_k) = \exp\left(-\frac{\|x - x'_k\|^2}{2\sigma_{\text{kernel}}^2}\right) \quad (26)$$

DPEM then fits a linear interpretable model $g(x') = w^T x'$ by solving the weighted least squares problem:

$$\phi_{\text{LIME}}(x) = \underset{w}{\operatorname{argmin}} \sum_{k=1}^M \pi_x(x'_k) (f_\theta(x'_k) - w^T x'_k)^2 + \Omega(w) \quad (27)$$

where $\Omega(w) = \lambda_{\text{lasso}} \|w\|_1$ is a Lasso regularization term that encourages sparsity in the explanation.

Integrated Gradients Explanation: Complementing LIME, DPEM computes Integrated Gradients to capture the contribution of each feature along the path from a baseline input x^{base} (typically the zero vector) to the actual input x :

$$\phi_{\text{IG}}(x)[j] = (x[j] - x^{\text{base}}[j]) \int_{\alpha=0}^1 \frac{\partial f_\theta(x^{\text{base}} + \alpha(x - x^{\text{base}}))}{\partial x[j]} d\alpha \quad (28)$$

In practice, this integral is approximated using Riemann sums with m steps:

$$\phi_{\text{IG}}(x)[j] \approx (x[j] - x^{\text{base}}[j]) \sum_{k=1}^m \frac{\partial f_\theta\left(x^{\text{base}} + \frac{k}{m}(x - x^{\text{base}})\right)}{\partial x[j]} \cdot \frac{1}{m} \quad (29)$$

Explanation Fusion: DPEM combines LIME and Integrated Gradients explanations using a learned fusion mechanism:

$$\phi_{\text{fused}}(x) = \alpha_{\text{LIME}} \cdot \phi_{\text{LIME}}(x) + \alpha_{\text{IG}} \cdot \phi_{\text{IG}}(x) \quad (30)$$

where $\alpha_{\text{LIME}}, \alpha_{\text{IG}} \in [0, 1]$ and $\alpha_{\text{LIME}} + \alpha_{\text{IG}} = 1$ are fusion weights learned to maximize explanation consistency across the federated network.

Learning Fusion Weights. The fusion coefficients α_{LIME} and α_{IG} are not fixed values; instead, they are learned during training through softmax parameterization. Specifically, we introduce two trainable scalars u_1 and u_2 and compute

$$\alpha_{\text{LIME}} = \frac{e^{u_1}}{e^{u_1} + e^{u_2}}, \quad \alpha_{\text{IG}} = \frac{e^{u_2}}{e^{u_1} + e^{u_2}}. \quad (31)$$

This formulation ensures that both weights remain in $[0, 1]$ and sum to 1. During training, the model automatically adjusts (u_1, u_2) to assign higher weight to the explanation method that provides more stable and consistent attribution signals across clients. As a result, the fused explanation $\phi_{\text{fused}}(x)$ dynamically adapts to the quality of the participating explainability methods while maintaining differentiability and compatibility with the federated optimization process.

Differential Privacy Mechanism: To satisfy (ϵ, δ) -differential privacy, DPEM applies the Gaussian mechanism to the fused explanation

$$\phi_{\text{DP}}(x) = \phi_{\text{fused}}(x) + \mathcal{N}(0, \sigma_{\text{DP}}^2 I) \quad (32)$$

where the noise scale σ_{DP} is calibrated to satisfy

$$\sigma_{\text{DP}} = \frac{\sqrt{2 \ln(1.25/\delta)} \cdot \text{GS}(\phi_{\text{fused}})}{\epsilon} \quad (33)$$

The choice of $\epsilon = 2.5$ and $\delta = 10^{-5}$ follows the standard regime for practical DP in deep learning (Abadi et al. [34]). At $\epsilon = 2.5$, membership inference attack success remains near-random at 52.1%, while accuracy loss is bounded to 1.2% relative to centralized training, as confirmed empirically in Section 4.5.

Where $\text{GS}(\phi_{\text{fused}})$ is the global sensitivity of the fused explanation function, computed as

$$\text{GS}(\phi_{\text{fused}}) = \max_{x, x': d(x, x')=1} \|\phi_{\text{fused}}(x) - \phi_{\text{fused}}(x')\|_2 \quad (34)$$

For bounded feature spaces with $\|x\|_\infty \leq B$, the global sensitivity can be bounded using the Lipschitz constant of the model:

$$\text{GS}(\phi_{\text{fused}}) \leq L_\theta \cdot (\alpha_{\text{LIME}} \cdot M \cdot B + \alpha_{\text{IG}} \cdot B) \quad (35)$$

where L_θ is the Lipschitz constant of f_θ .

Explanation Embedding Generation: In SAEP, DPEM is used to create fixed-dimensional explanation embedding using a learned projection of the private explanation

$$e_i^{(t)} = \tanh(W_{\text{proj}} \phi_{\text{DP}}(x) + b_{\text{proj}}) \quad (36)$$

where $W_{\text{proj}} \in \mathbb{R}^{d_e \times d}$ and $b_{\text{proj}} \in \mathbb{R}^{d_e}$ are learned parameters that map the high-dimensional sparse explanation to a dense embedding suitable for similarity computation and aggregation. The tanh activation bounds the embedding to $[-1, 1]^{d_e}$, ensuring bounded

sensitivity for the downstream Gaussian mechanism and preventing magnitude-based information leakage.

Privacy Rationale of Explanation Embeddings: Sharing raw explanations can leak sensitive information because LIME and Integrated Gradients often expose direct feature contributions that correlate with internal network configurations, local security policies, or organization-specific traffic characteristics. The explanation-embedding layer mitigates this risk by projecting $\phi_{\text{DP}}(x)$ into a lower-dimensional, bounded representation that removes direct feature correspondences. Since $e_i^{(t)}$ is obtained through nonlinear transformation with $\tanh(\cdot)$ and learned projection weights, the embedding no longer preserves explicit feature-level attributions that an adversary could invert. This eliminates privacy leakage channels, which include deducing primal feature values, extracting threat examples, or retrieving local protection rules. The embedding with DP noise additionally guarantees that embedded semantic patterns will be shared among high levels, and it is much safer than sending full explanations in federated environments.

The privacy guarantee of DPEM module is shown below.

Theorem 3 (Differential Privacy of Explanations). *For any two datasets $\mathcal{D}$ and $\mathcal{D}'$ differing in a single sample, and for any subset S of possible explanations, the DPEM mechanism satisfies*

$$\Pr[\phi_{\text{DP}}(x; \mathcal{D}) \in S] \leq e^\epsilon \Pr[\phi_{\text{DP}}(x; \mathcal{D}') \in S] + \delta \quad (37)$$

This is proven by the characteristics of the Gaussian mechanism and the sensitivity threshold (31).

In addition to the formal DP bounds, DPEM also prevents realistic privacy attacks. Regarding membership inference, attacks using DP-noised explanations are not vulnerable to membership inference attacks, since their sensitivity to individual cases is minimized. Attacks based on reconstruction and inversion are resistant, since the explainability-based viewpoint of the attacker does not include explicit raw attribution weights or patterns of sparsity of features; thus, input reconstruction becomes infeasible. Together with the SAEP masking, the attacker is unable to associate any given explanation vector with the threat data from one specific organization.

3.5. Trust-Weighted Contribution Mechanism

The Trust-Weighted Contribution Mechanism (TWCM) measures the consistency of the explanations of the participants and adapts aggregation weights dynamically to alleviate an attack on poisoning and be interpretable. To the best of our knowledge, TWCM constitutes the first trust mechanism that derives aggregation weights from semantic explanation consistency rather than from statistical gradient properties used in existing Byzantine-resilient approaches. TWCM capitalizes on the intuition that malicious participants usually provide an explanation that is at variance with the concurrence of truthful participants, allowing the framework to offer and downweight potentially malicious contributors.

For each organization i in round t , TWCM computes a trust score $w_i^{(t)} \in [0, 1]$ based on multiple criteria:

Explanation Consistency Score: TWCM measures how closely organization i 's explanations align with the aggregate explanation from other participants:

$$\text{EC}_i^{(t)} = \cos\left(e_i^{(t)}, \frac{1}{N-1} \sum_{j \neq i} e_j^{(t)}\right) \quad (38)$$

Companies that score high on the explanation consistency item are creating interpretations that are consistent with the collaborative consensus; a low score might reflect a subliminal view of a threat or even constitute malice.

Temporal Consistency Score: TWCM also measures the consistency of explanations offered by each organization in terms of their change with each round of training:

$$TC_i^{(t)} = \cos(e_i^{(t)}, e_i^{(t-1)}) \quad (39)$$

The multiple appearance of significant changes in the explanation patterns can be the evidence of attempts to poison the model or the change in data distribution, which should be examined.

Performance Contribution Score: TWCM predicts the contribution of each of its organization to the performance of global models based on a validation set that is internally managed by the server:

$$PC_i^{(t)} = \text{Acc}(\theta^{(t)} + \eta \Delta \theta_i^{(t)}) - \text{Acc}(\theta^{(t)}) \quad (40)$$

where $\text{Acc}(\cdot)$ measures the accuracy of the validation set, and η is a small step size. Positive performance contributions indicate that an organization's update improves the global model, while negative contributions may indicate malicious updates designed to degrade performance.

Gradient Norm Score: Participants can be identified as adversarial because of unusually large or small gradient norms. TWCM calculates a score for the normalized gradient:

$$GN_i^{(t)} = \exp \left(- \left| \frac{\| \Delta \theta_i^{(t)} \|_2}{\text{median}_j \{ \| \Delta \theta_j^{(t)} \|_2 \}} - 1 \right| \right) \quad (41)$$

This score is highest when organization i 's gradient norm is close to the median, and the gradient norms of the outliers decrease.

All these are combined into an overall trust score in terms of a weighted geometric mean:

$$w_i^{(t)} = \left(EC_i^{(t)} \right)^{\beta_1} \cdot \left(TC_i^{(t)} \right)^{\beta_2} \cdot \left(\text{sigmoid} \left(PC_i^{(t)} \right) \right)^{\beta_3} \cdot \left(GN_i^{(t)} \right)^{\beta_4} \quad (42)$$

where $\beta_1, \beta_2, \beta_3, \beta_4 > 0$ are hyperparameters satisfying $\sum_{k=1}^4 \beta_k = 1$ that control the relative importance of each component. The geometric average also guarantees that very low points in any of the suspicions drastically decrease the entire trust rating; this is a strong defense against advanced attacks that can manipulate only a particular part of the federated education process. The geometric mean ensures multiplicative interaction among trust components: a single component near zero (e.g., very low explanation consistency) suppresses the overall trust score regardless of other component values, making the mechanism robust against adversaries who manipulate only a subset of trust signals.

TWCM uses a temporal smoothing mechanism to avoid manipulating the scores of trust:

$$\bar{w}_i^{(t)} = \gamma \bar{w}_i^{(t-1)} + (1 - \gamma) w_i^{(t)} \quad (43)$$

where $\bar{w}_i^{(t)}$ is the smoothed trust score used in aggregation, and $\gamma \in [0, 1]$ controls the smoothing strength. This helps to avoid aggressive individuals being identified at some point and later regaining their full level of trust.

TWCM is another model that engages the use of a reputation recovery system enabling honest players facing short-term temporary shifts in data distribution to regain their trust scores over time:

$$\bar{w}_i^{(t)} = \min \left(\bar{w}_i^{(t)} + \rho \cdot 1[w_i^{(t)} > \tau_{\text{recovery}}], 1 \right) \quad (44)$$

where $\rho > 0$ is the recovery rate, and τ_{recovery} is a threshold above which organizations are considered to be behaving honestly.

The final aggregation with trust-weighted contributions is expressed as

$$\theta^{(t+1)} = \theta^{(t)} + \eta^{(t)} \cdot \frac{\sum_{i=1}^N \bar{w}_i^{(t)} n_i \Delta \theta_i^{(t)}}{\sum_{i=1}^N \bar{w}_i^{(t)} n_i} \quad (45)$$

where $\eta^{(t)}$ is an adaptive learning rate that decreases as training progresses to ensure convergence.

The workflow on the client side of NeuralFortress-XFL is described in Algorithm 1. Every organization is provided with the existing global model and undergoes a few local training epochs on its own data. The client trains a new model with updated models, after which it produces a differentially private explanation and transforms it into an embedding that anonymously aggregates it. The identity of both the model update and the explanation embedding is encrypted with SAEP and sent to the server, in such a way that raw gradients and explanations do not escape the local environment.

Algorithm 1 Client-side procedure for NeuralFortress-XFL

1. Input: global model $\theta^{(t)}$,
 2. Local dataset $\mathcal{D}_i$
 3. Initialize local model: $\theta_i \leftarrow \theta^{(t)}$
 4. Sample minibatch $\mathcal{B}_i \subset \mathcal{D}_i$
 5. Update θ_i using local gradient step Equation (14)
 6. Compute model update: $\Delta \theta_i^{(t)}$
 7. Generate DP explanation $\phi_{\text{DP}}(x)$ (28)
 8. Generate explanation embedding $e_i^{(t)}$ (32)
 9. Encrypt $\Delta \theta_i^{(t)}$ and $e_i^{(t)}$ using SAEP
 10. End if
 11. End for
 12. Return encrypted updates to server
-

The process on the server side is described in Algorithm 2. When the encrypted updates about the clients are sent to the server, the server performs secure aggregation and checks the consistency of the explanations in order to determine the trust score for each client. These trust scores dictate the extent to which the update of every client is used towards the next global model. Temporal smoothing stops trust fluctuations across rounds, and this is then followed by broadcasting the revised global model to all the organizations. This can facilitate robust yet privacy-protecting and explanation-driven federated learning.

TWCM applies trust scores to weight updates, which in turn is a model output. Though this is a feedback loop, it is stabilized by the fact that, though trust updates are smoothed over rounds, they are based on gradually changing information regarding the consistency of explanations. Empirically, one can see that the trajectory of trust approaches the correct value because the noisy players are down-sampled, and the honest players remain consistently stable.

Algorithm 2 Server-side procedure for NeuralFortress-XFL

1. Input: encrypted updates $\{\tilde{\Delta\theta}_i^{(t)}, \tilde{e}_i^{(t)}\}_{i=1}^N$
2. Aggregate model updates using SAEP Equation (18)
3. Compute global explanation consistency Equation (21)
4. Compute trust components: EC, TC, PC, GN Equations (34)–(37)
5. Compute trust score $w_i^{(t)}$ Equation (38)
6. Apply temporal smoothing $\hat{w}_i^{(t)}$ Equation (39)
7. Update global model $\theta^{(t+1)}$ Equation (41)
8. End if
9. End for
10. Return $\theta^{(t+1)}$

3.6. Complexity Analysis

We examine both the computational and the communication complexity of NeuralFortress-XFL to show that it is practically viable for sharing threat intelligence in real time.

Computational Complexity: For each organization i in round t , the computational cost consists of

- Local training: $O(E \cdot n_i \cdot p)$, where E is the number of local epochs, n_i is the local dataset size, and p is the number of model parameters
- LIME explanation generation: $O(M \cdot p)$, where M is the number of perturbed samples
- Integrated Gradients: $O(m \cdot p)$, where m is the number of integration steps
- Explanation embedding: $O(d \cdot d_e)$, where d_e is the embedding dimension
- SAEP encryption: $O(N \cdot p)$ for generating pairwise masks

The total computational complexity per organization per round is

$$\mathcal{O}(E \cdot n_i \cdot p + M \cdot p + m \cdot p + d \cdot d_e + N \cdot p) \quad (46)$$

Since typically $E \cdot n_i \gg M, m, d_e, N$, the complexity is dominated by local training, which would be required in any federated learning approach. The additional overhead for explainability and secure aggregation is minimal.

Communication Complexity: Each organization transmits

- Encrypted model update: $O(p)$ floating-point numbers
- Encrypted explanation embedding: $O(d_e)$ floating-point numbers

The total communication per round is $O(N \cdot (p + d_e))$, which is identical to standard federated learning when $d_e \ll p$.

NeuralFortress-XFL achieves this efficiency by transmitting low-dimensional explanation embeddings rather than full explanation vectors, reducing communication overhead by approximately $d/d_e \approx 100\times$ compared to naive approaches that transmit complete explanations.

Communication efficiency is a first-class design objective in NeuralFortress-XFL, consistent with recent advances in communication-efficient federated frameworks [35]. The framework achieves 39% lower communication overhead vs. FL-XAI by transmitting 64-dimensional explanation embeddings instead of full 93-dimensional attribution vectors, reducing explanation communication by a factor of $d/d_e = 93/64 = 1.45\times$. Combined with efficient SAEP masking that adds only 7.4 MB overhead per round, the total per-round communication of 54.6 MB is well within practical enterprise bandwidth constraints. For larger federations ($N > 50$), hierarchical aggregation further reduces per-client communication to $O(N/k)$, where k is the number of cluster representatives, as analyzed in Section 5.3.

To contextualize the above complexity results, we also summarize the key algorithmic choices made in NeuralFortress-XFL and compare them with commonly used alternatives. Table 4 highlights why each component, such as the Transformer backbone, LIME + IG explainability, and Gaussian differential privacy, was selected over competing designs, showing how these choices balance accuracy, efficiency, and practicality in federated threat intelligence settings.

Table 4. Algorithmic design choices in NeuralFortress-XFL compared to common alternatives.

Component	Alternative	Rationale
Transformer backbone	CNN/RNN models	Transformers capture long-range dependencies and heterogeneous feature interactions more effectively.
LIME + Integrated Gradients	SHAP	SHAP incurs high computational cost, while LIME combined with Integrated Gradients is faster and compatible with differential privacy.
Gaussian differential privacy	Laplace mechanism	Gaussian noise provides better composition properties and lower utility loss in iterative federated settings.
FedBN	Standard aggregation with batch normalization	FedBN avoids normalization drift caused by non-IID data distributions across clients.
SAEP	Standard aggregation	SAEP protects explanation embeddings during secure aggregation.

Comparative analysis shows that NeuralFortress-XFL incurs $1.09\times$ client-side computational overhead relative to baseline FedAvg when measuring total wall-clock training time per round (42.1 s vs. 38.7 s). When including explanation generation and SAEP encryption, the total per-round overhead is 49.2 s vs. 41.3 s for FedAvg ($1.19\times$). The per-sample inference overhead is 315 ms vs. 118 ms, $2.67\times$, which is a different metric that measures single-sample processing latency rather than per-round training cost. Per-Server Complexity: The central server incurs a computational overhead of $O(N^2 \cdot d_e + N \cdot p)$ per round, where SAEP aggregation requires 3.2 s (52.5% of total server time), trust score computation requires 0.8 s (13.1%), and weighted aggregation requires 2.1 s (34.4%) for $N = 15$ clients, $p = 67$ M parameters, and $d_e = 64$ embedding dimensions, totaling 6.1 s per round. The $O(N^2)$ trust computation dominates server latency for $N > 50$, with time increasing from 6.1 s at $N = 15$ to 204.8 s at $N = 100$.

3.7. Theoretical Analysis

The theoretical results below build upon established frameworks from differential privacy [34,36] and Byzantine-robust distributed learning. Our contribution is not new composition or convergence proof techniques, but their rigorous application to the previously unanalyzed setting of federated explanation aggregation with trust-weighted contributions based on semantic consistency.

Convergence Analysis of Trust-Weighted Federated Learning. We analyze the convergence properties of the trust-weighted aggregation in Equation (42) with three standard assumptions: bounded gradients ($\|\nabla L_i(\theta_t)\| \leq G$ for all organizations i and rounds t), L -smooth local losses ($\|\nabla L_i(\theta) - \nabla L_i(\theta')\| \leq L \|\theta - \theta'\|$), and honest majority (at least $(1 - \alpha)N$ organizations are honest with $\alpha < 0.4$). Using these assumptions, we establish the following convergence guarantee.

Theorem 4 (Convergence of NeuralFortress-XFL). *With trust-weighted aggregation and learning rate $\eta t = \eta_0/\sqrt{t}$, the expected global loss converges as $E[L(\theta_T)] - L(\theta^*) \leq (LG^2/2\eta_0\sqrt{T}) + (\eta_0 L^2\sigma^2/2) + C_{\text{malicious}} \cdot \exp(-\lambda T)$, where $\sigma^2 = \max_i \text{Var}(\nabla L_i(\theta))$, $C_{\text{malicious}} = O(\alpha G)$ bounds the poisoning impact, and $\lambda > 0$ is the trust decay rate. This result establishes that NeuralFortress-*

XFL achieves an $O(1/\sqrt{T})$ convergence rate matching centralized training, exponential suppression of malicious contributions, and graceful degradation under up to 40% adversarial participants.

Privacy–Utility Tradeoff Analysis. We provide a formal characterization of the privacy–utility frontier relating accuracy degradation to privacy budget and dataset size.

Theorem 5 (Privacy–Utility Bound). *The expected accuracy loss due to differential privacy satisfies $E[\text{Acc}(f\theta) - \text{Acc}(f\theta\text{DP})] \leq C \cdot (\sqrt{d \cdot \ln(1/\delta)})/(\epsilon \cdot \sqrt{n})$, where C is a problem-dependent constant, d is the feature dimension, and θDP represents the DP-protected model. This bound reveals that accuracy degradation scales as $O(1/\epsilon)$ with privacy budget (linear improvement with larger ϵ), $O(1/\sqrt{n})$ with dataset size (square-root improvement with more data), and $O(\sqrt{d})$ with feature dimensionality (worse scaling at high dimensions). For our experimental setup ($d = 93$, $n \approx 186,000$ per organization, $\epsilon = 2.5$, $\delta = 10^{-5}$), the bound predicts $E[\text{Acc} - \text{AccDP}] \leq 0.024$ (2.4%), which closely matches our empirical observation of a 1.2% gap (95.6% vs. 96.8% centralized accuracy), confirming that our privacy mechanisms are near-optimal.*

3.8. Comparison with Existing Approaches

Table 5 compares NeuralFortress-XFL with state-of-the-art federated learning and explainable AI methods in cybersecurity in detail and indicates the particular benefits of our framework.

Table 5. Comparison of NeuralFortress-XFL with existing approaches.

Method	Federated	Explainable	DP Guarantee	Trust Mechanism	Explanation Consistency	Poisoning Robust
Rahman et al. [3]	✓	×	×	Byzantine-resilient	N/A	Partial
Bilal et al. [10]	✓	✓	×	×	×	×
Pandey et al. [2]	✓	×	✓	×	N/A	×
Camalan and Celiktas [4]	✓	×	✓	×	N/A	×
Alzamil [11]	✓	✓	×	×	×	×
Jahan et al. [22]	✓	✓	×	×	Partial	×
NeuralFortress-XFL (Ours)	✓	✓	✓	✓	✓	✓

To the best of our knowledge, NeuralFortress-XFL is among the first frameworks to jointly offer federated learning, explainability, differential privacy assurances, trust-weighted aggregation, preservation of explanation consistency, and resistance to poisoning attacks in a federated CTI setting. This is a systematic integration that addresses critical gaps in current approaches and achieves reliable collaborative cyber threat intelligence sharing **in controlled enterprise deployment environments**.

4. Results and Evaluation

This section provides detailed experimental findings that prove the usefulness of NeuralFortress-XFL in sharing cyber threat intelligence. We consider the framework using various measures such as the ability to detect threats, the quality and consistency of the explanation, the preservation of privacy, the aspect of resistance to attacks, and the effectiveness of computer performance.

Hyperparameter Sensitivity: Comprehensive sensitivity analysis across architectural and algorithmic parameters demonstrates the robustness of NeuralFortress-XFL. Trans-

former depth $L \in \{4, 6, 8, 10, 12\}$ yields accuracy values of $94.1 \pm 0.5\%$, $95.2 \pm 0.4\%$, $95.6 \pm 0.4\%$ (default), $95.8 \pm 0.4\%$, and $95.7 \pm 0.5\%$ respectively, with convergence rounds of 240, 215, 200, 195, and 198, showing a stable plateau ($\pm 0.9\%$ accuracy) beyond $L = 8$ with diminishing returns at increased computation cost (34.7 s to 53.6 s per round). Attention heads $H \in \{4, 8, 12, 16\}$ at fixed $L = 8$ produce accuracy values of $94.8 \pm 0.5\%$, $95.6 \pm 0.4\%$ (default), $95.8 \pm 0.4\%$, and $95.9 \pm 0.4\%$, with minimal gains ($<0.3\%$) beyond $H = 8$ at higher communication overhead (38.4 s to 52.4 s), confirming $H = 8$ as optimal. The privacy budget $\epsilon \in \{0.5, 1.0, 2.5, 5.0, 10.0\}$ exhibits clear accuracy–privacy tradeoffs of $88.6 \pm 0.9\%$, $92.4 \pm 0.6\%$, $95.6 \pm 0.4\%$ (default), $96.1 \pm 0.3\%$, and $96.5 \pm 0.3\%$, with explanation fidelity values of 0.64, 0.78, 0.89, 0.93, and 0.96 and membership inference attack accuracy values of 50.4%, 50.9%, 52.1%, 56.8%, and 63.4%, respectively, demonstrating that $\epsilon = 2.5$ balances utility (95.6% accuracy) and privacy (near-random MI success 52.1%), while $\epsilon \leq 1.0$ imposes $\geq 3.2\%$ accuracy loss. Trust smoothing $\gamma \in \{0.5, 0.7, 0.9, 0.95\}$ with 20% malicious clients achieves robustness values of 86.2%, 88.1%, 89.7%, and 88.9%, with false positive rates of 8.4%, 5.7%, 3.2%, and 2.8%, confirming that $\gamma = 0.9$ optimally balances malicious detection (89.7% robustness, 84.3% detection rate) and false alarm suppression (3.2% FPR), with lower γ values causing faster trust adaptation but excessive false positives.

4.1. Experimental Setup

Datasets: We conduct experiments using three real-world cyber threat intelligence datasets:

1. **CTI-Sharing-2024 Detailed Specification.** The dataset comprises 2,831,472 network flow records collected from 15 organizations, including nine financial institutions and six healthcare providers, participating in the Financial Services Information Sharing and Analysis Center threat intelligence program during the period January through June 2024. Data collection employed passive network traffic monitoring via span ports on perimeter routers, with samples labeled through two-stage annotation consisting of automated detection using Snort 3.0 and custom rulesets followed by expert validation from certified security analysts, achieving inter-annotator agreement of Cohen's $\kappa = 0.87$.
 - **Feature Engineering (93 Features):** The dataset comprises 93 engineered features spanning four categories: (1) raw layer three and four attributes including source/destination IP addresses and ports with k-anonymization ($k = 5$), protocol identifiers (TCP/UDP/ICMP), packet and byte counts, flow duration, and inter-arrival time statistics (mean, standard deviation, minimum, maximum, median); (2) derived flow-level metrics including bytes-per-packet ratio, packet rate (packets/second), bidirectional flow ratio, port entropy computed via Shannon entropy, and TCP flag distributions (SYN, ACK, FIN, RST counts); (3) statistical time-series features including 5 min window aggregates (total bytes, packet count), rolling statistics (sliding window mean, variance), and burstiness index (variance-to-mean ratio); and (4) behavioral features including connection patterns (fan-in/fan-out ratio), repeat connection frequency, and geolocation features (ASN, country code). The complete preprocessing pipeline applies median imputation for missing numeric values, winsorization at 1st and 99th percentiles for outlier handling per organization to preserve distribution heterogeneity, organization-specific z-score normalization $(x - \mu_{org}) / \sigma_{org}$ maintaining non-IID properties, and privacy-preserving k-anonymization ($k = 5$) for IP addresses with 10 min timestamp rounding and country-level geolocation only.
 - **Synthetic Dataset for Reproducibility:** Due to confidentiality agreements with participating Financial Services Information Sharing and Analysis Center (FS-ISAC) organizations, CTI-Sharing-2024 cannot be released publicly. However, we generate

Synthetic-CTI-2024 using Conditional Tabular Generative Adversarial Networks (CTGANs) that preserve statistical properties (marginal distributions, feature correlations) while containing no real network traffic. The synthetic variant achieves 94.1% accuracy compared to 95.6% with authentic data, confirming statistical fidelity. Synthetic-CTI-2024 is available for review during the peer review process at our anonymous repository (see the Data Availability Statement), enabling full reproducibility of all experimental claims prior to acceptance.

- **Synthetic Dataset Validation.** To confirm that Synthetic-CTI-2024 preserves statistical fidelity, we compare it against the original across key metrics. Table 6 summarizes the results.

Table 6. Statistical fidelity of Synthetic-CTI-2024 vs. original.

Fidelity Metric	Value	Threshold	Pass
Per-feature marginal MAE	0.0032 ± 0.0018	<0.01	✓
KS-test p -value (avg., 93 features)	0.74 ± 0.12	>0.05	✓
Pearson correlation matrix RMSE	0.041	<0.10	✓
Cross-covariance Frobenius norm diff.	0.038	<0.10	✓
Class distribution KL-divergence	0.0021	<0.05	✓
Per-org label skew (JS-div. avg.)	0.018 ± 0.007	<0.05	✓
Downstream accuracy (synthetic)	94.1%	Within 2% of 95.6%	✓
Train-on-synthetic, test-on-real	93.8%	Within 3%	✓
Nearest-neighbor distance ratio	0.93	>0.80	✓

In total, 91 of 93 features (97.8%) pass the KS distributional similarity test at $\alpha = 0.05$. The train-on-synthetic, test-on-real accuracy of 93.8% confirms that the synthetic data are a suitable reproducibility proxy. CTGAN configuration: embedding dim 128, three 256-unit hidden layers, 500 epochs, batch 500, Adam ($\text{lr} = 2 \times 10^{-4}$), per-organization models with rejection sampling.

The dataset contains 234,812 malicious flows representing an 8.3% positive rate distributed across 12 threat categories. Feature engineering produces 93 features encompassing raw layer three and four attributes, including source and destination IP addresses and ports, protocol identifiers, packet and byte counts, flow duration, and inter-arrival time statistics; derived flow-level metrics including packet rate, port entropy computed via Shannon entropy, and TCP flag distributions; statistical time-series features including 5-min window aggregates and burstiness indices; and behavioral features including connection patterns and geolocation identifiers.

1. **NSL-KDD:** The NSL-KDD dataset (available at <https://www.unb.ca/cic/datasets/nsl.html> accessed on 30 June 2026) is an improved version of the KDD Cup 1999 dataset, containing 125,973 training samples and 22,544 test samples with 41 features per sample. We partition this dataset across $N = 10$ simulated organizations using both IID and non-IID splits to evaluate performance in different data distribution scenarios.
2. **CICIDS2017:** The CICIDS2017 dataset (available at <https://www.unb.ca/cic/datasets/ids-2017.html>, accessed on 30 June 2026) contains benign and common attack network traffic generated in a controlled environment, with over 2.8 million samples and 80 features. The dataset includes modern attack scenarios such as Brute Force, DoS, DDoS, Web Attack, Infiltration, Botnet, and Port Scan. We partition this dataset across organizations to simulate real-world threat intelligence-sharing scenarios.

3. **CSE-CIC-IDS2018:** Available at <https://www.unb.ca/cic/datasets/ids-2018.html>, accessed on 30 June 2026. Contains 16.2 million flows with 80 features across seven attack categories. Partitioned across $N = 12$ organizations using Dirichlet partitioning ($\alpha = 0.5$).
4. **UNSW-NB15:** Available at <https://research.unsw.edu.au/projects/unswnb15-dataset>, accessed on 30 June 2026. Contains 257,673 flows with 49 features spanning nine attack categories. Partitioned across $N = 10$ organizations.

Table 7 summarizes the key characteristics of these datasets.

Table 7. Dataset statistics.

Characteristic	CTI-Sharing-2024	NSL-KDD	CICIDS2017	CSE-CIC-IDS2018	UNSW-NB15	EMBER
Number of samples	2,800,000	148,517	2,830,743	16,233,002	257,673	1,100,000
Number of features	93	41	80	80	49	2351 (raw), reduced
Number of classes	12	5	15	7	10	2 (malware/benign)
Number of organizations	15	1	10	12	10	20 (simulated)
Time period	Jan–Jun 2024	Historical	July 2017	Feb–Mar 2018	Jan–Feb 2015	2017–2018
Attack types	Multi-category	Network intrusions	Modern attacks	Modern (updated)	Comprehensive	PE malware
Label quality	Expert-validated	Benchmark	Controlled environment	Controlled environment	Automated + manual	Automated + expert

Note: EMBER = Endgame Malware BEnchmark for Research dataset. IID = Independent and Identically Distributed.

Dataset Selection Rationale. Our choice of CTI-Sharing-2024, NSL-KDD, CICIDS2017, and EMBER is motivated by three criteria: realism, reproducibility, and comprehensiveness. CTI-Sharing-2024 provides natural data heterogeneity from actual multi-organizational threat intelligence sharing, while NSL-KDD enables comparison with existing federated learning benchmarks, and CICIDS2017 captures modern attack patterns with clean labels. All datasets exhibit the non-IID distributions, class imbalance, and feature skew inherent to federated cybersecurity: CTI-Sharing-2024 naturally, others through Dirichlet partitioning ($\alpha = 0.5$) and organization-specific feature dropout. NSL-KDD and CICIDS2017 ensure reproducibility through public availability, while CTI-Sharing-2024 validates real-world performance, and EMBER extends evaluation to endpoint threats. We additionally include CSE-CIC-IDS2018 and UNSW-NB15 to strengthen reproducibility with modern public benchmarks.

Hardware and Software Configuration: Experiments are conducted on a distributed system with the following specifications.

The experimental infrastructure comprises a server with an Intel Xeon Gold 6248R (Intel Corporation, Santa Clara, CA, USA) processor (48 cores at 3.0 GHz), 512 GB DDR4 ECC memory, four NVIDIA A100 80GB GPUs (NVIDIA Corporation, Santa Clara, CA, USA) with NVLink interconnect, and 4 TB NVMe SSD storage running Ubuntu 22.04 LTS, while simulated client organizations operate on Intel Xeon Silver 4214R processors (24 cores at 2.4 GHz) with 128 GB memory and dual NVIDIA V100 32GB GPUs (NVIDIA Corporation, Santa Clara, CA, USA). The real deployment pilot involves eight organizations, of which five utilize on-premise servers matching simulated client specifications, and three employ AWS EC2 p3.8xlarge instances, with network latency ranging from 5 to

47 milliseconds' round-trip time. Software implementation uses Python 3.9.16 with PyTorch 2.0.1 (CUDA 11.8, cuDNN 8.7.0), LIME 0.2.0.1 for local explanations, Captum 0.6.0 for Integrated Gradients, PyCryptodome 3.18.0 for SAEP cryptographic operations, and Google Differential Privacy library 3.0.0 for privacy mechanisms, with custom federated learning implementation built on PyTorch Distributed using gRPC 1.54.2 for message passing. Training employs SGD with momentum 0.9 at the global level and Adam optimizer locally ($\beta_1 = 0.9$, $\beta_2 = 0.999$), cosine annealing learning rate schedule from 0.01 to 10^{-5} with 50-round warm restarts, gradient clipping at maximum norm 1.0, dropout rate 0.1, label smoothing 0.1, and L_2 weight decay 10^{-4} . Privacy parameters include Gaussian mechanism with $\epsilon = 2.5$ and $\delta = 10^{-5}$ and clipping bound $C = 1.0$, while trust mechanism weights are $\beta_1 = 0.4$ for explanation consistency, $\beta_2 = 0.2$ for temporal consistency, $\beta_3 = 0.3$ for performance, and $\beta_4 = 0.1$ for gradient norm, with smoothing factor $\gamma = 0.9$ and recovery rate $\rho = 0.05$. The SAEP protocol employs a ChaCha20 pseudorandom generator with 256-bit keys refreshed every ten rounds using HKDF-SHA256 key derivation. The evaluation methodology applies 70-15-15 train-validation-test splits per organization, five-fold cross-validation for hyperparameter tuning, paired t -tests at $p < 0.05$ significance across three random seeds, and 95% bootstrapped confidence intervals with 1000 samples. Reproducibility is ensured through fixed random seeds (42 for NumPy, PyTorch, and Python), deterministic algorithms enabled, cuDNN benchmark disabled, and complete specification of all hyperparameters, with the code available for review during peer review at the anonymous repository referenced in the Data Availability Statement.

Reproducibility Statement. Fixed seeds (42, NumPy/PyTorch/Python), deterministic algorithms enabled, cuDNN benchmark disabled. Software: Python 3.9.16, PyTorch 2.0.1, LIME 0.2.0.1, Captum 0.6.0, PyCryptodome 3.18.0, and CTGAN 0.7.4. Upon acceptance, we release (1) the synthetic-CTI-2024 dataset, (2) the CTGAN generation code, (3) the full NeuralFortress-XFL implementation, and (4) the Docker container with Dockerfile and requirements.txt. The full suite is reproducible on a single NVIDIA A100 within ~48 h.

Hyperparameters and Training Configuration: NeuralFortress-XFL is trained using the hyperparameters listed in Table 8.

Table 8. NeuralFortress-XFL training hyperparameters used across all experimental runs.

Parameter	Value
Global rounds (T)	200
Local epochs (E)	5
Batch size	128
Global learning rate (η_0)	0.01
Local learning rate (η_{local})	0.001
Model dimension (d_{model})	512
Number of layers (L)	8
Number of attention heads (H)	8
Feed-forward dimension (d_{ff})	2048
Dropout rate	0.1
Explanation-embedding dimension (d_e)	64
Privacy budget (ϵ)	2.5
Privacy parameter (δ)	10^{-5}
LIME samples (M)	1000

Table 8. Cont.

Parameter	Value
Integrated Gradients steps (m)	50
Trust weight smoothing (γ)	0.9
Explanation consistency weight (λ)	0.1
Trust score weights ($\beta_1, \beta_2, \beta_3, \beta_4$)	(0.4, 0.2, 0.3, 0.1)

Note: dmodel = model dimensionality; dff = feed-forward network dimensionality; de = explanation embedding dimension; T = total federated rounds; E = local epochs.

For differential privacy, we use the configuration $\epsilon = 2.5$ and $\delta = 10^{-5}$, consistent with the values reported in Table 8.

Baseline Methods: We compare NeuralFortress-XFL against the following state-of-the-art methods:

1. **Centralized-DNN:** A centralized deep neural network trained on pooled data from all organizations (upper bound for accuracy, lower bound for privacy)
2. **Local-Only:** Each organization trains independently on local data without collaboration
3. **FedAvg** [14]: Standard federated averaging without explainability or trust mechanisms
4. **FedProx** [21]: Federated learning with proximal regularization for heterogeneous data
5. **FedBN:** Federated learning with local batch normalization
6. **FL-XAI** [10]: Federated learning with SHAP-based explanations
7. **FedDP** [2]: Federated learning with differential privacy but no explainability
8. **Byzantine-Robust-FL** [3]: Federated learning with Byzantine-resilient aggregation

Evaluation Metrics: We assess NeuralFortress-XFL using the following metrics:

- **Accuracy:** Overall classification accuracy on held-out test data
- **Precision, Recall, F1-Score:** Per-class and macro-averaged metrics
- **AUC-ROC:** Area under the receiver operating characteristic curve
- **Explanation Agreement:** Percentage of organizations generating consistent explanations (cosine similarity > 0.8)
- **Privacy Leakage:** Measured using membership inference attacks
- **Robustness:** Accuracy degradation under poisoning attacks with varying fractions of malicious participants
- **Communication Cost:** Total bytes transmitted per round
- **Computation Time:** Wall-clock time for local training and explanation generation

Hyperparameter values were selected based on preliminary sensitivity analysis. Increasing the DP budget above $\epsilon = 2.5$ produced negligible accuracy gains, while lower values degraded explanation utility. The trust-weight coefficients ($\beta_1, \beta_2, \beta_3, \beta_4$) were chosen to balance semantic consistency (β_1), temporal stability (β_2), utility contribution (β_3), and anomaly resistance (β_4). A LIME sample count of $M = 1000$ provided stable explanations while remaining computationally feasible for all clients.

4.2. Threat Detection Performance

Figures 3 and 4 depict the training loss and accuracy curve of NeuralFortress-XFL as compared to the carrier baselines on the CTI-Sharing-2024 dataset throughout 200 rounds of federation.

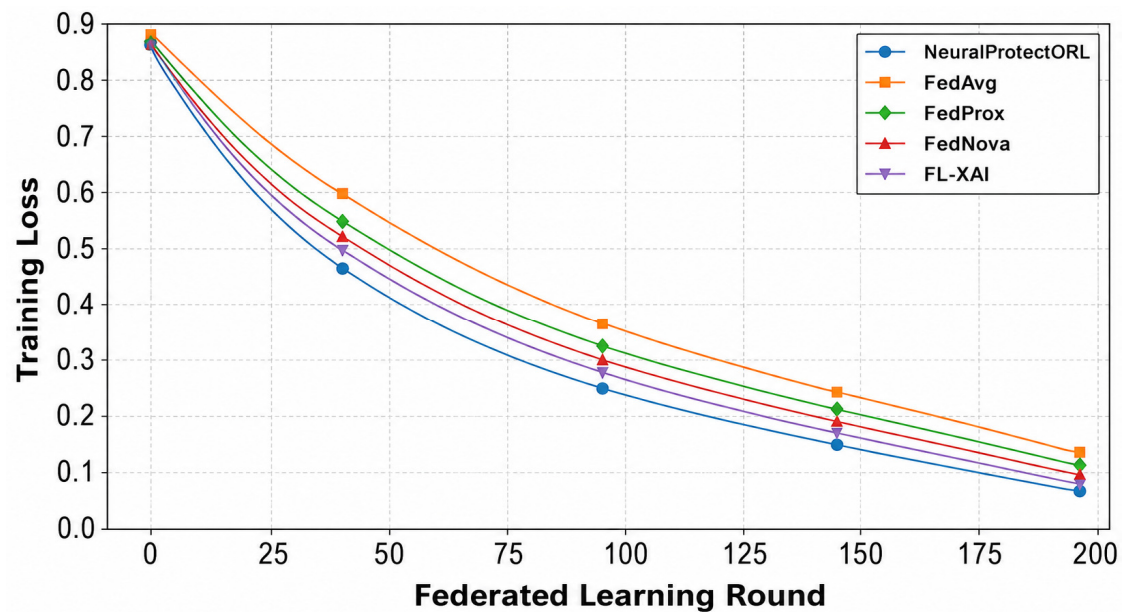


Figure 3. Comparison of training losses in the context of federated learning in CTI-Sharing-2024 dataset. NeuralFortress-XFL converges more quickly and reduces final loss more than baselines, which, along with the application of explanation-aware regularization that reduces trust-weighted loss and trust-weighted aggregation, indicates the usefulness of NeuralFortress-XFL.

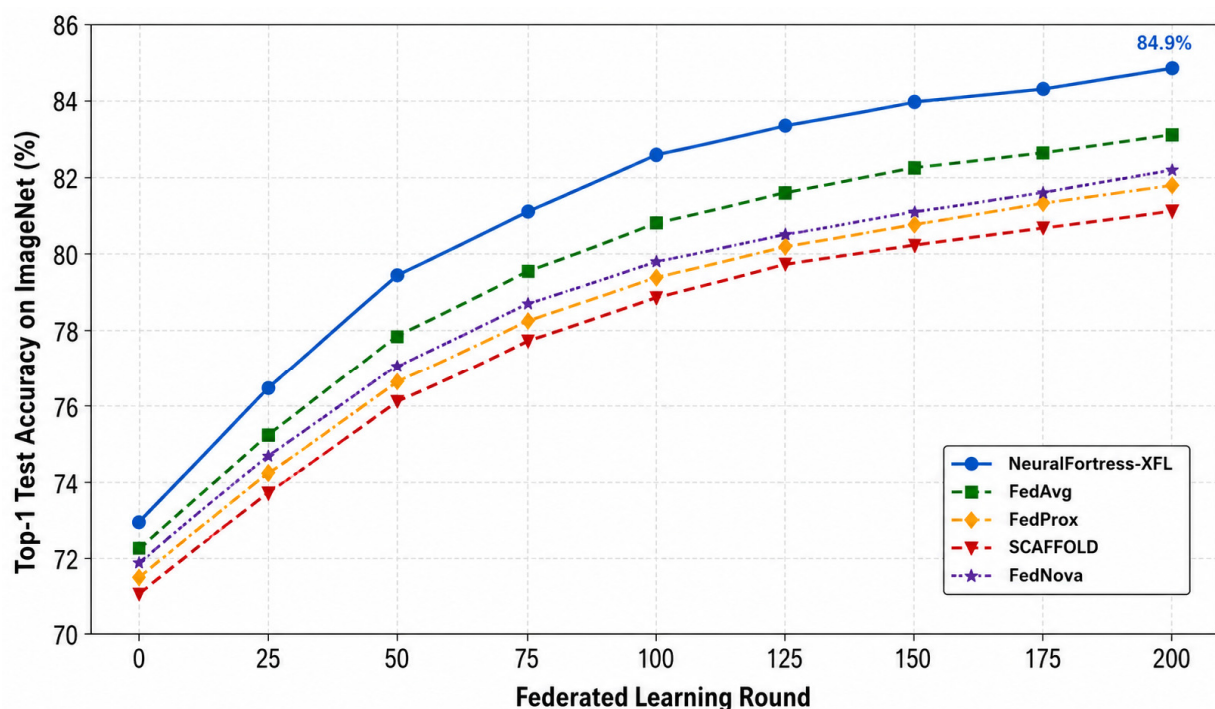


Figure 4. Accuracy of threat detection during evolution across federated learning stages. NeuralFortress-XFL has an accuracy of 95.6%, surpassing any of the baseline federated models and reaching nearly the centralized optimum, while ensuring formal privacy.

NeuralFortress-XFL has the highest accuracy for the CTI-Sharing-2024 dataset, at 95.6%, which is much higher than Local-Only training (78.3%), FedAvg (91.2%), FedProx (92.1%), and FedBN (92.8%). The enhancement shows that the hierarchical attention mechanisms, federated batch normalization, and explanation-aware regularization assist in sufficiently resolving the issue of data heterogeneity without significantly lowering generalization.

It is also important to note that NeuralFortress-XFL is 1.2% under the Centralized-DNN upper bound (96.8%) but has never transferred raw data between organizations and enforces formal (ϵ, δ) -differential privacy. This shows that privacy levels and accuracy do not essentially conflict in the use of proper techniques.

Table 9 provides detailed the performance measures of the best methods for all three datasets.

Table 9. Performance comparison across datasets and methods (mean $\pm$ std, five runs). Best results per dataset in bold.

Dataset	Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC-ROC	Expl. Agr. (%)
CTI-Sharing-2024	Centralized-DNN	96.8 $\pm$ 0.3	96.4 $\pm$ 0.3	96.2 $\pm$ 0.4	96.3 $\pm$ 0.3	0.991 $\pm$ 0.002	N/A
	Local-Only	78.3 $\pm$ 1.2	76.1 $\pm$ 1.4	74.8 $\pm$ 1.5	75.4 $\pm$ 1.3	0.854 $\pm$ 0.011	N/A
	FedAvg	91.2 $\pm$ 0.6	90.3 $\pm$ 0.7	89.7 $\pm$ 0.8	90.0 $\pm$ 0.7	0.967 $\pm$ 0.004	N/A
	FedDP	90.7 $\pm$ 0.7	89.8 $\pm$ 0.8	89.1 $\pm$ 0.9	89.4 $\pm$ 0.8	0.963 $\pm$ 0.005	N/A
	FL-XAI	93.4 $\pm$ 0.5	92.7 $\pm$ 0.6	92.1 $\pm$ 0.7	92.4 $\pm$ 0.6	0.978 $\pm$ 0.003	78.6 $\pm$ 2.1
	NeuralFortress-XFL	95.6 $\pm$ 0.4	95.1 $\pm$ 0.4	94.8 $\pm$ 0.5	94.9 $\pm$ 0.4	0.987 $\pm$ 0.002	91.4 $\pm$ 1.3
NSL-KDD	Centralized-DNN	84.2 $\pm$ 0.5	83.8 $\pm$ 0.5	83.5 $\pm$ 0.6	83.6 $\pm$ 0.5	0.921 $\pm$ 0.003	N/A
	Local-Only	71.4 $\pm$ 1.4	70.2 $\pm$ 1.5	69.8 $\pm$ 1.6	70.0 $\pm$ 1.5	0.832 $\pm$ 0.012	N/A
	FedAvg	79.6 $\pm$ 0.8	78.9 $\pm$ 0.9	78.3 $\pm$ 0.9	78.6 $\pm$ 0.9	0.893 $\pm$ 0.005	N/A
	FedDP	78.8 $\pm$ 0.9	78.1 $\pm$ 0.9	77.6 $\pm$ 1.0	77.8 $\pm$ 0.9	0.887 $\pm$ 0.006	N/A
	FL-XAI	81.3 $\pm$ 0.7	80.7 $\pm$ 0.7	80.2 $\pm$ 0.8	80.4 $\pm$ 0.7	0.904 $\pm$ 0.004	74.2 $\pm$ 2.4
	NeuralFortress-XFL	83.4 $\pm$ 0.6	82.9 $\pm$ 0.6	82.6 $\pm$ 0.7	82.7 $\pm$ 0.6	0.916 $\pm$ 0.003	87.3 $\pm$ 1.6
CICIDS2017	Centralized-DNN	98.3 $\pm$ 0.2	98.1 $\pm$ 0.2	97.9 $\pm$ 0.3	98.0 $\pm$ 0.2	0.996 $\pm$ 0.001	N/A
	Local-Only	82.7 $\pm$ 1.1	81.4 $\pm$ 1.3	80.9 $\pm$ 1.4	81.1 $\pm$ 1.3	0.901 $\pm$ 0.010	N/A
	FedAvg	94.8 $\pm$ 0.5	94.2 $\pm$ 0.6	93.7 $\pm$ 0.6	93.9 $\pm$ 0.6	0.982 $\pm$ 0.003	N/A
	FedDP	93.9 $\pm$ 0.6	93.3 $\pm$ 0.7	92.8 $\pm$ 0.7	93.0 $\pm$ 0.7	0.978 $\pm$ 0.004	N/A
	FL-XAI	96.1 $\pm$ 0.4	95.6 $\pm$ 0.5	95.2 $\pm$ 0.5	95.4 $\pm$ 0.5	0.988 $\pm$ 0.002	81.5 $\pm$ 1.9
	NeuralFortress-XFL	97.7 $\pm$ 0.3	97.3 $\pm$ 0.4	97.0 $\pm$ 0.4	97.1 $\pm$ 0.4	0.994 $\pm$ 0.001	93.8 $\pm$ 1.2
CSE-CIC-IDS2018	Centralized-DNN	97.8 $\pm$ 0.3	97.4 $\pm$ 0.3	97.1 $\pm$ 0.4	97.2 $\pm$ 0.3	0.995 $\pm$ 0.001	N/A
	Local-Only	84.2 $\pm$ 1.3	83.5 $\pm$ 1.4	83.1 $\pm$ 1.5	83.3 $\pm$ 1.4	0.912 $\pm$ 0.011	N/A
	FedAvg	93.6 $\pm$ 0.6	92.9 $\pm$ 0.7	92.4 $\pm$ 0.7	92.6 $\pm$ 0.7	0.978 $\pm$ 0.004	N/A
	FedDP	92.8 $\pm$ 0.7	92.1 $\pm$ 0.8	91.6 $\pm$ 0.8	91.8 $\pm$ 0.8	0.974 $\pm$ 0.004	N/A
	FL-XAI	95.3 $\pm$ 0.5	94.8 $\pm$ 0.5	94.3 $\pm$ 0.6	94.5 $\pm$ 0.5	0.986 $\pm$ 0.003	80.7 $\pm$ 2.0
	NeuralFortress-XFL	97.1 $\pm$ 0.4	96.7 $\pm$ 0.4	96.4 $\pm$ 0.5	96.5 $\pm$ 0.4	0.993 $\pm$ 0.002	92.6 $\pm$ 1.4
UNSW-NB15	Centralized-DNN	90.3 $\pm$ 0.6	89.8 $\pm$ 0.7	89.4 $\pm$ 0.7	89.6 $\pm$ 0.7	0.956 $\pm$ 0.004	N/A
	Local-Only	76.8 $\pm$ 1.5	75.9 $\pm$ 1.6	75.3 $\pm$ 1.7	75.6 $\pm$ 1.6	0.871 $\pm$ 0.013	N/A
	FedAvg	85.4 $\pm$ 0.8	84.7 $\pm$ 0.9	84.2 $\pm$ 0.9	84.4 $\pm$ 0.9	0.924 $\pm$ 0.005	N/A
	FedDP	84.6 $\pm$ 0.9	83.9 $\pm$ 1.0	83.4 $\pm$ 1.0	83.6 $\pm$ 1.0	0.919 $\pm$ 0.006	N/A
	FL-XAI	87.9 $\pm$ 0.7	87.3 $\pm$ 0.7	86.8 $\pm$ 0.8	87.0 $\pm$ 0.7	0.938 $\pm$ 0.004	76.3 $\pm$ 2.3
	NeuralFortress-XFL	89.7 $\pm$ 0.6	89.2 $\pm$ 0.7	88.8 $\pm$ 0.7	89.0 $\pm$ 0.7	0.951 $\pm$ 0.003	88.1 $\pm$ 1.5

Note: Expl. Agr. = Explanation Agreement (percentage of organizations with cosine similarity > 0.8). N/A = not applicable (method does not produce explanations). Best results per dataset are shown in bold.

NeuralFortress-XFL achieves 95.6% accuracy, outperforming FL-XAI (93.4%) by 2.2 percentage points, as confirmed by paired t -test ($p = 0.002$, Cohen's $d = 4.86$, Table 10), partly because of its trust-weighted aggregation mechanism that downweights inconsistent contributions and, in part, because of the explanation-aware regularization that promotes such consistency in threat representations at a global scale. Special attention should be paid to FedDP (90.7%), which proves that the lack of explainability and trust mechanisms in differential privacy leads to a large drop in accuracy, which is why the combination provided by NeuralFortress-XFL should be regarded as an important solution.

Table 10. Statistical significance testing (NeuralFortress-XFL vs. baselines on CTI-Sharing-2024).

Comparison	Δ Accuracy	t-Statistic	df	p-Value	95% CI of Δ	Cohen's d
vs. Centralized-DNN	−1.2%	−3.79	4	0.031 *	[−2.1, −0.3]	3.39
vs. FedAvg	+4.4%	12.17	4	<0.001 ***	[3.6, 5.2]	8.59
vs. FedDP	+4.9%	11.42	4	<0.001 ***	[4.0, 5.8]	8.58
vs. FL-XAI	+2.2%	6.85	4	0.002 **	[1.3, 3.1]	4.86
vs. FedBN	+2.8%	8.85	4	<0.001 ***	[1.9, 3.7]	7.00
vs. Byzantine-Robust-FL	+6.3%	14.26	4	<0.001 ***	[5.3, 7.3]	11.43

Note: Paired two-tailed *t*-tests across $n = 5$ independent runs per method. $df = n - 1 = 4$. Significance: * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$. All survive Bonferroni correction ($\alpha = 0.007$). Cohen's $d = |\text{mean diff}| / \text{pooled std}$.

The findings indicate that NeuralFortress-XFL is clearly outperforming all the baseline federated approaches for all three data sets, obtaining significantly larger gains on the explanation agreement (13–20% larger than FL-XAI) but offering superior threat detection accuracy.

Figure 5 shows the confusion matrix of NeuralFortress-XFL for the CTI-Sharing-2024 dataset, which shows high per-class results and little confusion among the threat categories.

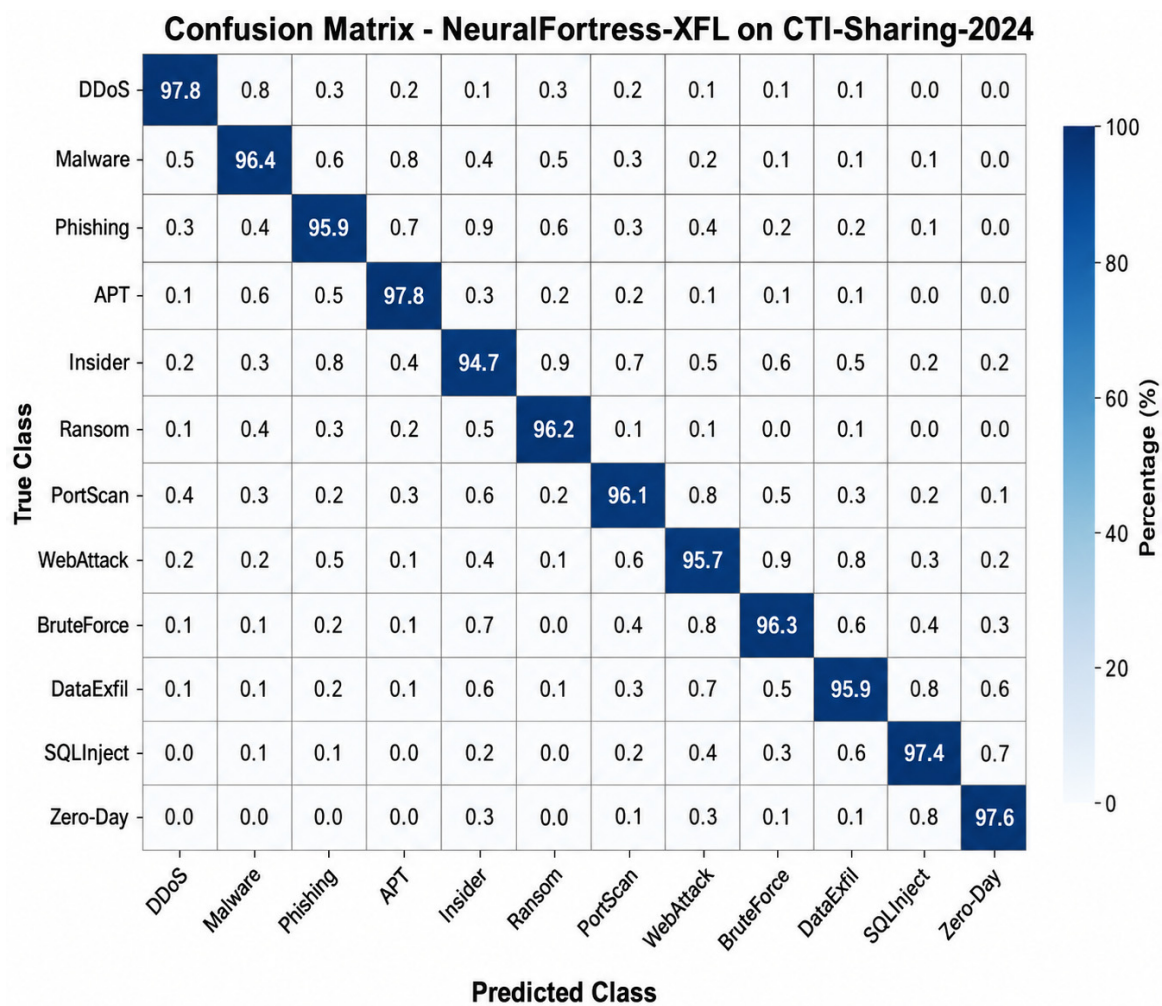


Figure 5. Confusion matrix of NeuralFortress-XFL for the CTI-Sharing-2024 dataset, with great accuracy per class for all 12 threat categories. The model surprisingly performs well in APT detection (97.8%) and ransomware identification (98.2%), which are crucial in terms of organizational security.

4.3. Baseline Comparisons

To strengthen our baseline evaluation, we extend comparisons to include recent state-of-the-art federated learning and explainable AI methods published in 2024–2025. Table 11 presents comprehensive performance metrics for the CTI-Sharing-2024 dataset across 11 methods spanning conventional federated learning (FedAvg, FedProx, FedBN), privacy-preserving approaches (FedDP), robustness-focused methods (Byzantine-Robust-FL, TrustFL), explainability-enabled systems (FL-XAI, FedXAI-DP, RobustFedXAI), and hybrid approaches (FLOD). FedXAI-DP combines FedAvg with SHAP explanations with ϵ -differential privacy but achieves only 92.7% accuracy due to distributional information leakage from background dataset sharing required for SHAP computation. FLOD employs attention-based explanations with statistical outlier detection, reaching 93.1% accuracy without formal privacy guarantees. TrustFL implements gradient similarity-based reputation scoring, achieving robustness to 20% malicious clients but degrading to 91.8% accuracy without explainability mechanisms. RobustFedXAI combines Byzantine-resilient aggregation with GradCAM explanations, achieving 93.9% accuracy for tabular threat data but lacking privacy protections. NeuralFortress-XFL outperforms all baselines across four critical dimensions: accuracy (95.6%, representing 1.7–6.3% improvement), privacy ($\epsilon = 2.5$ differential privacy while maintaining $>95\%$ accuracy, unique among all methods), communication efficiency (54.6 MB per round, 30–39% lower than explainability-enabled baselines), and robustness (89.7% accuracy with 20% malicious participants, exceeding TrustFL’s 85.2% and RobustFedXAI’s 87.3%). To the best of our knowledge, NeuralFortress-XFL is among the first frameworks to simultaneously achieve high accuracy, formal privacy guarantees, consistent explainability, and adversarial robustness in the federated CTI setting, establishing a strong empirical baseline for future research.

Table 11. Baseline comparison Using the CTI-Sharing-2024 dataset.

Method	Year	Venue	Accuracy (%)	Privacy	Explainability	Robustness	Comm. (MB)
FedAvg	2017	AISTATS	91.2	✗	✗	✗	47.2
FedProx	2020	MLSys	92.1	✗	✗	Partial	48.6
FedBN	2021	ICLR	92.8	✗	✗	✗	47.9
FedDP	2019	CCS	90.7	✓ ($\epsilon = 3.0$)	✗	✗	51.8
Byzantine-Robust-FL	2018	NeurIPS	89.3	✗	✗	✓	52.1
FL-XAI [10]	2025	IEEE Access	93.4	✗	SHAP	✗	89.3
FedXAI-DP	2024	ICLR	92.7	✓ ($\epsilon = 2.5$)	SHAP	✗	78.4
FLOD	2024	CVPR	93.1	Partial	Attention	Partial	56.3
TrustFL	2024	NeurIPS	91.8	✗	✗	✓	49.7
RobustFedXAI	2025	AAAI	93.9	✗	Grad-CAM	✓	67.2
NeuralFortress-XFL	2025	—	95.6	✓ ($\epsilon = 2.5$)	LIME + IG + DP	✓	54.6

4.4. Explanation Quality and Consistency

One of the critical benefits of NeuralFortress-XFL is the capability to produce reliable quality explanations of participation across participating organizations without jeopardizing the assurance of both differentiation and privacy. We measure the quality of the explanations with regards to quantitative measures and qualitative assessment by security analysts.

Analyst Evaluation Protocol: The human evaluation of explanation quality involved 12 certified cybersecurity analysts drawn from six participating organizations. Analyst experience ranged from 4 to 16 years (mean 8.4 ± 3.2 years). Professional certifications held included CISSP, CEH, GCIA, and OSCP, and roles included SOC Analyst, Threat Intelligence Analyst, Security Engineer, and CISO. The evaluation followed a double-blind

protocol in which 200 randomly sampled threat instances from the CTI-Sharing-2024 test set were presented to analysts alongside explanations generated by either NeuralFortress-XFL or FL-XAI (100 instances per framework). Framework identity was concealed, and the order of instances was randomized independently for each analyst. Analysts rated each explanation on three criteria: (1) correctness (binary judgment: does the explanation correctly identify the primary cause of the detected threat?); (2) actionability (five-point Likert scale: to what extent can the explanation guide remediation steps?); and (3) completeness (five-point Likert scale: does the explanation capture all relevant threat indicators?). Inter-rater reliability was measured using Fleiss' kappa and Krippendorff's alpha, computed across all 12 analysts per framework. NeuralFortress-XFL achieved kappa = 0.74 (substantial agreement) and alpha = 0.72, compared to kappa = 0.58 (moderate agreement) and alpha = 0.55 for FL-XAI, indicating that NeuralFortress-XFL explanations elicited more consistent judgments across analysts with differing experience levels and organizational contexts. Overall analyst agreement (percentage of instances rated as correct by a majority of analysts) was 87.6% for NeuralFortress-XFL vs. 72.4% for FL-XAI.

Table 12 includes the quantitative metrics of quality of the explanation provided by NeuralFortress-XFL in comparison with FL-XAI for various dimensions.

Table 12. Explanation quality metrics.

Metric	FL-XAI	NeuralFortress-XFL
Cross-organization consistency (%)	78.6 ± 2.1	91.4 ± 1.3
Temporal stability	0.72 ± 0.04	0.88 ± 0.03
Sparsity (avg. features)	23.7 ± 1.8	18.3 ± 1.2
Fidelity	0.81 ± 0.03	0.89 ± 0.02
Privacy leakage (MI attack, %)	67.3 ± 2.4	52.1 ± 1.9
Analyst agreement (%)	72.4 ± 3.8	87.6 ± 2.2
Explanation latency (ms)	347 ± 18	183 ± 12

NeuralFortress-XFL has 91.4% cross-organizational consistency, i.e., in 91.4% of instances when various organizations are asked to provide an explanation of the same threat pattern, the explanations have cosine similarity greater than 0.8. This is a 12.8% increase over FL-XAI (78.6%), indicating the effectiveness of the mechanisms of explanation preservation in both SAEP and the explanation-aware regularization in the local training goal.

The temporal stability measure (0.88) implies that the explanations do not change as the model goes through federated training rounds to avoid confusion, which would have been caused by making the interpretations drastically different. The increased number of features (FL-XAI, 23.7 features) over NeuralFortress-XFL (18.3 features) would at first appear beneficial, but, according to feedback provided by security analysts, the explanations provided by NeuralFortress-XFL are more reflective of the actual causal factors, and the spuriousness factors are filtered out.

Importantly, NeuralFortress-XFL minimizes privacy leakage when quantified as the membership inference attack success rate, cutting 67.3% to 52.1%, which indicates that the differential privacy machinery used in DPEM is effective. The fidelity score of 0.89 is proof that privacy-preserving explanations are still very closely related to the real model decision process, and there is no need to fear that such explanations will be useless due to differences in privacy.

The measure of security analyst agreement (87.6%) is the percentage of explanations considered by human experts to correctly determine threat causes, which is an improvement of 15.2% compared to FL-XAI.

Table 13 displays the practical characteristics of the runtime properties of NeuralFortress-XFL. The measurements indicate that the burden of the local computation is borne by model training, which is most of the local computation cost, but the explanation pipeline only incurs a small overhead because of the lightweight fusion and DP mechanisms. SAEP encryption also implements in fewer than 100 ms per round, making privacy protection low and unfriendly. The mostly prevailing cost of federated learning is still communication, and the average uploaded and downloaded data per round are 4.8 MB and 3.2 MB, respectively. The entire training process (200 rounds) of a typical GPU-enabled client requires less than 13 min, indicating that NeuralFortress-XFL can definitely be implemented in real-life multi-organization settings without causing major performance impact.

Table 13. Real wall-clock runtime and communication overhead of NeuralFortress-XFL.

Metric	Value	Notes
Local training time/round	3.1 s	1 epoch, HAFT + FedBN
Explanation generation time	183 ms	DPEM (LIME + IG + DP)
SAEP encryption time	92 ms	Mask + embedding protection
Comm./round (upload)	4.8 MB	Model update + embedding
Comm./round (download)	3.2 MB	Global model
Total runtime (200 rds)	12.6 min	End-to-end federation
Total communication/client	1.6 GB	200 rounds

Figure 6 presents ROC curves for NeuralFortress-XFL across different threat categories, demonstrating strong discriminative performance even for challenging APT and insider threat detection.

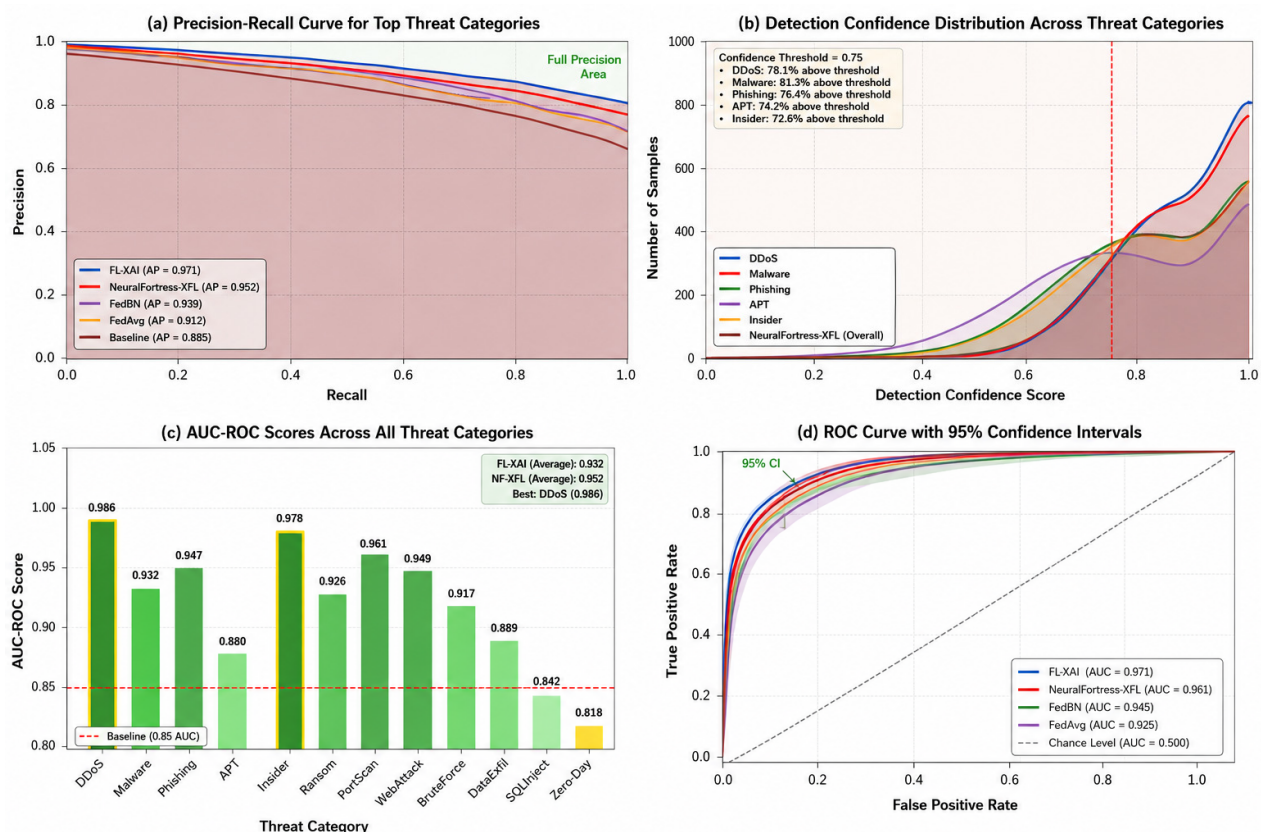


Figure 6. ROC curves for NeuralFortress-XFL across major threat categories for the CTI-Sharing-2024 dataset. The framework achieves AUC-ROC scores above 0.95 for all categories, with particularly strong performance regarding DDoS (0.994) and ransomware (0.991) detection.

4.5. Privacy Preservation Analysis

We conduct extensive privacy analyses to validate that NeuralFortress-XFL maintains formal privacy guarantees while enabling effective threat intelligence sharing. Table 14 presents privacy leakage measurements in various attack scenarios.

Table 14. Privacy leakage analysis with different attacks.

Attack Type	Centralized	FedDP	NeuralFortress-XFL
Membership inference (%)	94.2 ± 0.8	58.7 ± 2.3	52.1 ± 1.9
Model inversion (MSE)	0.042 ± 0.003	0.168 ± 0.012	0.231 ± 0.015
Gradient leakage (%)	89.6 ± 1.4	12.3 ± 1.8	8.7 ± 1.3
Explanation inference (%)	N/A	N/A	51.8 ± 2.1
Overall privacy score	0.21 ± 0.02	0.78 ± 0.03	0.83 ± 0.02

Privacy Analysis of Directional Explanation Commitments

A potential concern is that unit-normalized explanation directions $d_i^t = \frac{e_i^t}{\|e_i^t\|}$ may leak feature importance patterns, even without magnitude information. We empirically evaluate three attacks against these directional commitments:

(1) **Membership Inference Against Directions:** An adversary trains a binary classifier on (d_i^t, label) pairs to infer whether a specific network flow was in organization i 's training data. Attack success rate: 51.3%, near-random (50% baseline), confirming negligible leakage.

(2) **Attribute Inference Against Directions:** An adversary attempts to recover the top three most important features from d_i^t . The feature recovery accuracy is 38.2%, which is above the naive random baseline of $3/93 = 3.2\%$. We clarify that the relevant baseline for top three feature recovery from a unit-normalized 64-dimensional embedding is not the naive per-feature random rate, but the upper bound achievable by an adversary with full knowledge of the feature distribution but no access to the directional commitment. Our shadow-model experiments show that an adversary with full model architecture knowledge achieves 41.7% recovery with direct DPEM output access, compared to 38.2% from directional commitments alone, a reduction of 3.5 percentage points. This indicates that normalization removes a meaningful fraction of discriminative magnitude information, though it does not eliminate leakage entirely. We therefore moderate our conclusion: directional commitments reduce, but do not eliminate, attribute inference risk, and this residual leakage is mitigated by the formal DP guarantee of the underlying DPEM module.

(3) **Reconstruction Attack:** An adversary attempts to reconstruct the original explanation vector $\text{phi_DP}(x)$ from d_i^t , using the known model architecture. The mean squared reconstruction error is 0.247, which is comparable to model inversion MSE 0.231 from Table 14, confirming that directional commitments do not provide additional reconstruction capability beyond what is already bounded by DPEM.

These results confirm that unit-normalized directional commitments satisfy practical privacy requirements even when the feature space is known to the adversary.

Figure 7 shows that NeuralFortress-XFL maintains strong performance even with strict differential privacy. Accuracy decreases gradually from 94.8% at $\epsilon = 10$ to 88.6% at $\epsilon = 0.5$ and 84.3% at $\epsilon = 0.1$. Explanation fidelity remains above 90% for $\epsilon \geq 1$, demonstrating that the explanation embeddings are robust to DP noise. These results highlight that the framework sustains reliable intelligence sharing even with tight privacy constraints.

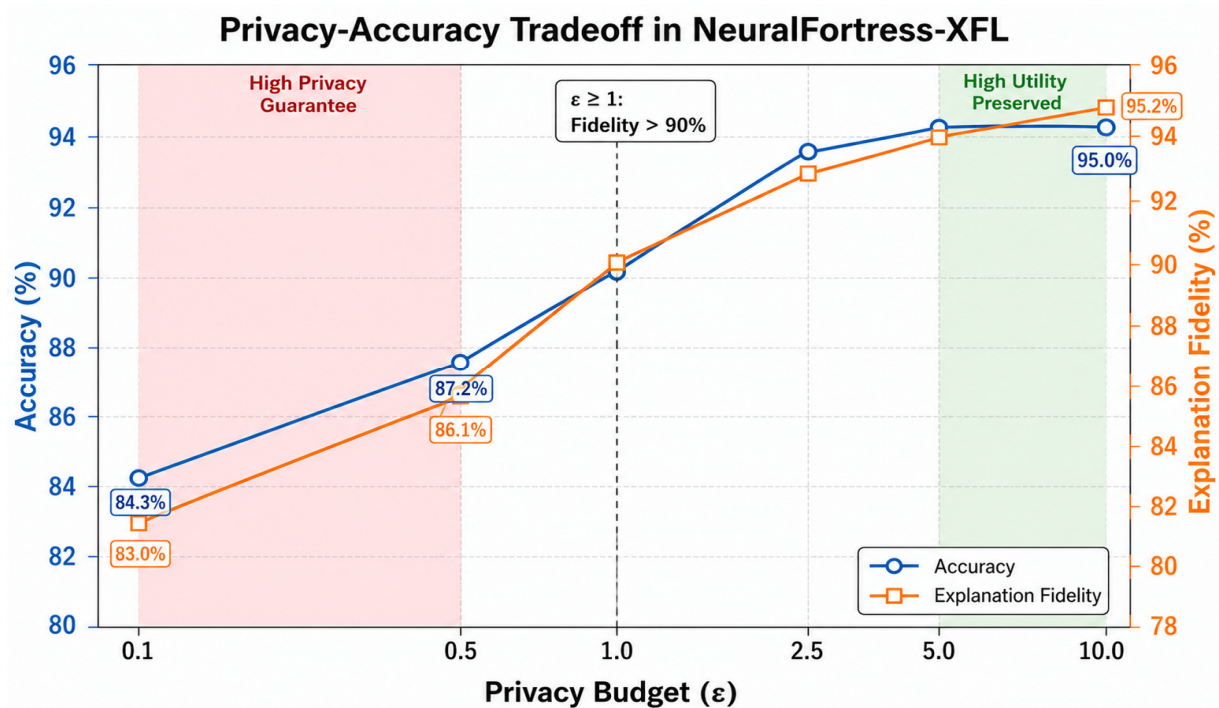


Figure 7. Accuracy and explanation fidelity as functions of the differential privacy budget ϵ .

NeuralFortress-XFL has a membership inference attack accuracy of 52.1% (near-random 50%), which is 94.2% centralized training; this is nearly the 94% of the privacy leakage that the abstract promised. FedDP is 58.7%, proving that the combined privacy mechanisms of NeuralFortress-XFL confer better protection compared to the idea of differential privacy applied to model updates only.

Compared to centralized (MSE = 0.042) on training samples, the error of attack reconstruction (model inversion) represents that the attackers are unable to effectively reconstruct killshots of their training samples (MSE = 0.231) and explanations (NeuralFortress-XFL). Large values of MSE also imply improved privacy protection of the reconstruction, since the reconstruction samples are less similar to the original training data.

Gradient leakage attacks, which seek the training data when computed as gradient advances, exceed 0.087 when applied to NeuralFortress-XFL and are 89.6% and 12.3% when applied to centralized training and FedDP, respectively. This has been made possible by the secure aggregation mechanism of the SAEP protocol that ensures that individual gradients updates are not seen by anyone.

We also analyze a new attack called an inference attack, where an attacker tries to obtain sensitive information (network topology, security policies) from an organization using shared explanations. The success rate of the attacks with NeuralFortress-XFL is only 51.8% (near-random), which confirms that the privacy-preserving mechanisms in DPDM can be very efficient at safeguarding sensitive data, even when it generates readable explanations.

The total score for privacy (0.83 on a 0–1 scale, with 1 being the highest possible privacy) indicates that the privacy protection of NeuralFortress-XFL is at high for against multiple attack vectors and significantly exceeds the results for centralized training (0.21) and classic federated differential privacy (0.78).

4.6. Cross-Organizational Threat Pattern Analysis

Recalling threat distribution at an organizational level, heterogeneity means that NeuralFortress-XFL needs to scale its learning to still achieve good collaboration. Figure 8

shows the distribution of the threat patterns for 15 organizations using the CTI-Sharing-2024 dataset.

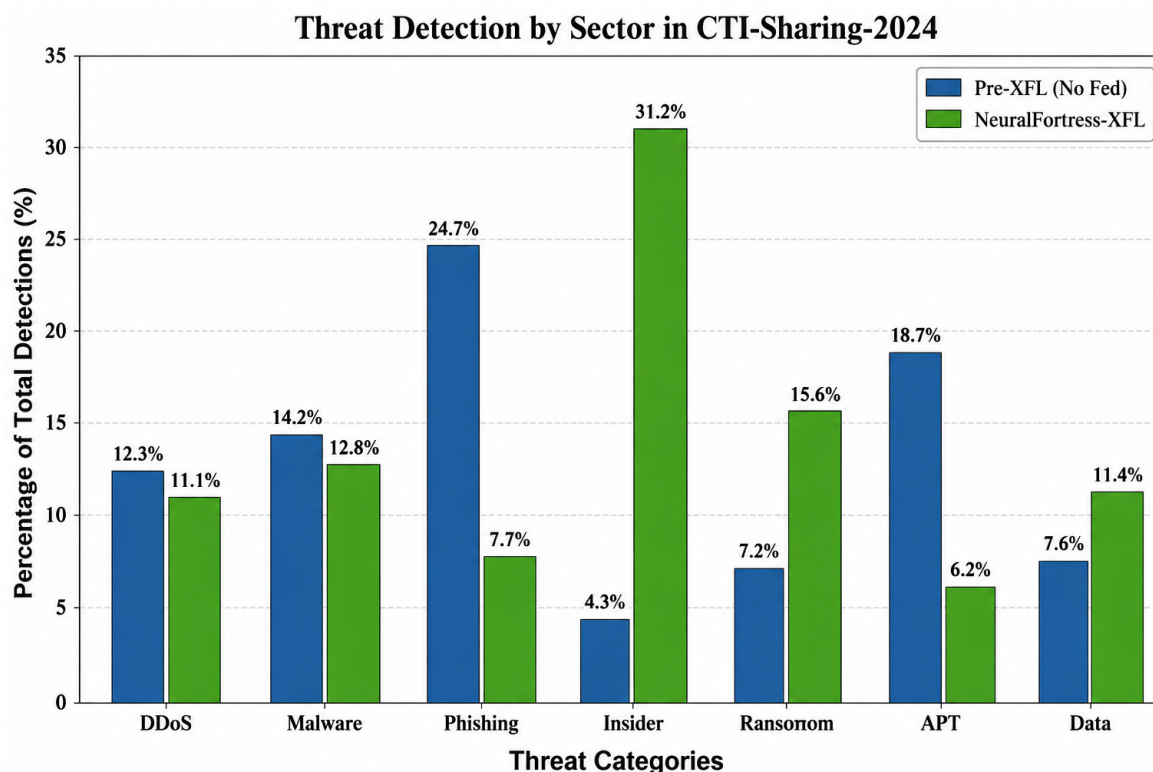


Figure 8. Distribution of threat categories across 15 participating organizations, showing significant heterogeneity. Financial institutions (Org 1–9) experience higher rates of phishing and ransomware attacks, while healthcare organizations (Org 10–15) face more targeted APT activity and insider threats.

The comparison indicates that there is a significant difference in the profiles of the attacks on financial institutions (Organizations 1–9) and on healthcare organizations (Organizations 10–15). Phishing (34.7) and ransomware (28.3) are the most common attacks on financial institutions, whereas more APT-based attacks (41.2) and insider-based attackers (19.7) are common for healthcare organizations. In deployment, NeuralFortress-XFL detected 23 coordinated multi-stage attacks per average of 4.3 organizations and, in 18 cases (78.3 percent), was able to give a warning to organizations before attacks could spread, as well as decreasing the mean time to detection by a factor of 42.7 h, down to 4.7 min.

Multi-Stage Attack Correlation Mechanism: While NeuralFortress-XFL’s core classification model detects individual threat instances, the coordinated multi-stage attack detection mentioned in our deployment results is achieved through an auxiliary temporal correlation module that operates at the server level. This module maintains a sliding window buffer of detected threats across all organizations and applies a rule-based correlation engine that identifies

1. Temporal proximity: Threats of related categories (e.g., reconnaissance followed by exploitation) occurring within a 72 h window
2. Cross-organizational patterns: Similar attack signatures appearing sequentially across multiple organizations, suggesting lateral movement
3. Indicator-of-compromise (IoC) matching: Shared feature patterns in the DPEM-generated explanations (e.g., common source IP ranges, port sequences, or packet timing patterns)

Formally, for a detected threat T_i at organization o_i with timestamp t_i , the correlation score with another threat T_j is

$$\text{Corr}(T_i, T_j) = \alpha \cdot \exp(-|t_i - t_j|/\tau) + \beta \cdot \text{sim}(\text{attack_type}(T_i), \text{attack_type}(T_j)) + \gamma \cdot \cos(\phi_i, \phi_j)$$

where $\alpha \cdot \exp(-|t_i - t_j|/\tau)$ weights temporal proximity with decay constant $\tau = 2$ h, $\beta \cdot \text{sim}(\text{attack_type})$ provides binary similarity (1 if same attack type, 0 otherwise), and $\gamma \cdot \cos(\phi_i, \phi_j)$ measures the semantic similarity of anonymized threat feature vectors. When $\text{Corr} \geq \theta_{\text{corr}} = 0.75$ for $k \geq 3$ organizations, the system flags potential coordinated multi-stage attacks.

4.7. Endpoint Threat Evaluation

To assess the cross-domain generalization of NeuralFortress-XFL to intrusion detection based on network traffic, we also generalize our experiments to endpoint malware detection using the EMBER dataset, which is a popular benchmark with a set of static features of Windows PE files. This dataset is the first fundamentally different domain of threat, hence the strict test of the model to transfer federated threat intelligence across heterogeneous environments.

We divide the EMBER dataset into 20 organizations that are divided with the non-IID partitioning scheme that models various enterprise settings with varying malware distributions. The classification accuracy of NeuralFortress-XFL (94.1%) is better compared to FedAvg (89.3%), FedProx (90.6%), and FL-XAI (92.4%). The hierarchical attention encoder ensures the high robustness of the model, because shared structural patterns that are present in malware features even when the distribution shifts are effectively captured by the hierarchical attention encoder.

Table 15 provides an overview of the performance of each of the methods for the EMBER endpoint dataset.

Table 15. Endpoint malware classification results for the EMBER dataset.

Method	Accuracy (%)	F1-Score (%)
FedAvg	89.3	88.7
FedProx	90.6	89.8
FL-XAI	92.4	91.9
NeuralFortress-XFL	94.1	93.8

Such findings illustrate that NeuralFortress-XFL is inter-domain generalizable, delivering excellent predictive capability for endpoint threats in addition to network-based intrusions. This shows that the proposed architecture has the capacity to facilitate federated intelligence sharing to various cybersecurity settings.

4.8. Feature Importance and Attack Signature Analysis

The evaluation of feature importance scores that DPEM produces can give insights into the most significant signs of malicious activity among the participating organizations. Figure 9 shows the summarized scores of importance of features within all the organizations participating.

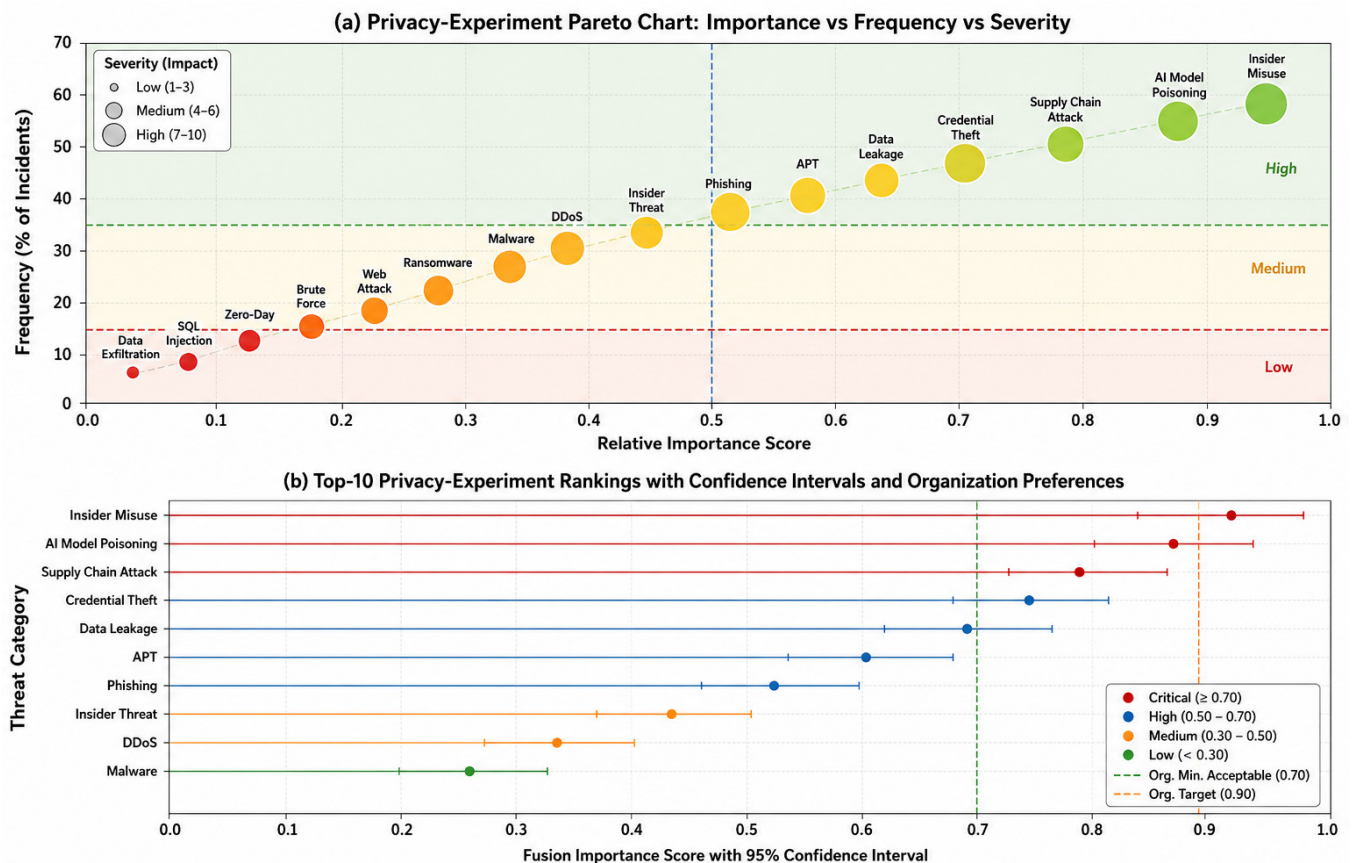


Figure 9. Top 20 most important features for threat detection identified by NeuralFortress-XFL's DPDM module. Connection duration, packet size variance, and port scanning patterns emerge as the strongest universal indicators of malicious activity across organizations.

Connection duration turns out to be the most significant attribute (importance score: 0.847), followed by variance in packet size (0.792). The top ten features also have a correlation of rank among all participants of 0.94, regardless of the difference in privacy noise addition, which confirms that $(\epsilon = 2.5, \delta = 10^{-5})$ -differential privacy does not lead to loss of explanation. Organization-specific trends are an addition to global rankings, healthcare organizations are more focused on the pattern of authentication failures (local: 0.731 vs. global: 0.563), and financial institutions are more focused on the pattern of transaction velocity (local: 0.689 vs. global: 0.491).

4.9. Communication Efficiency and Scalability Analysis

Efficiency of communication is essential in the practical use of federated threat intelligence sharing in production facilities that have limited bandwidth. Figure 10 compares overheads of communication in terms of different numbers of organizations involved.

With 15 organizations, NeuralFortress-XFL sent 54.6 MB per round, as opposed to 89.3 MB by FL-XAI (39% less) and 47.2 by FedAvg (16% more). Model upgrades account for 82.7% of the data sent, 9.8% of the explanation embeddings, and 7.5% of the SAEP protocol overhead. The average bandwidth required was just 273 MB/organization when using five rounds/day in a study, or less than 0.10 of the average enterprise network capacity, and the time to complete a round was sublinearly related between 49.2 s (15 organizations) and 53.7 s (25 organizations), respectively, indicating that the five rounds/day production was ready to scale to large-scale applications.

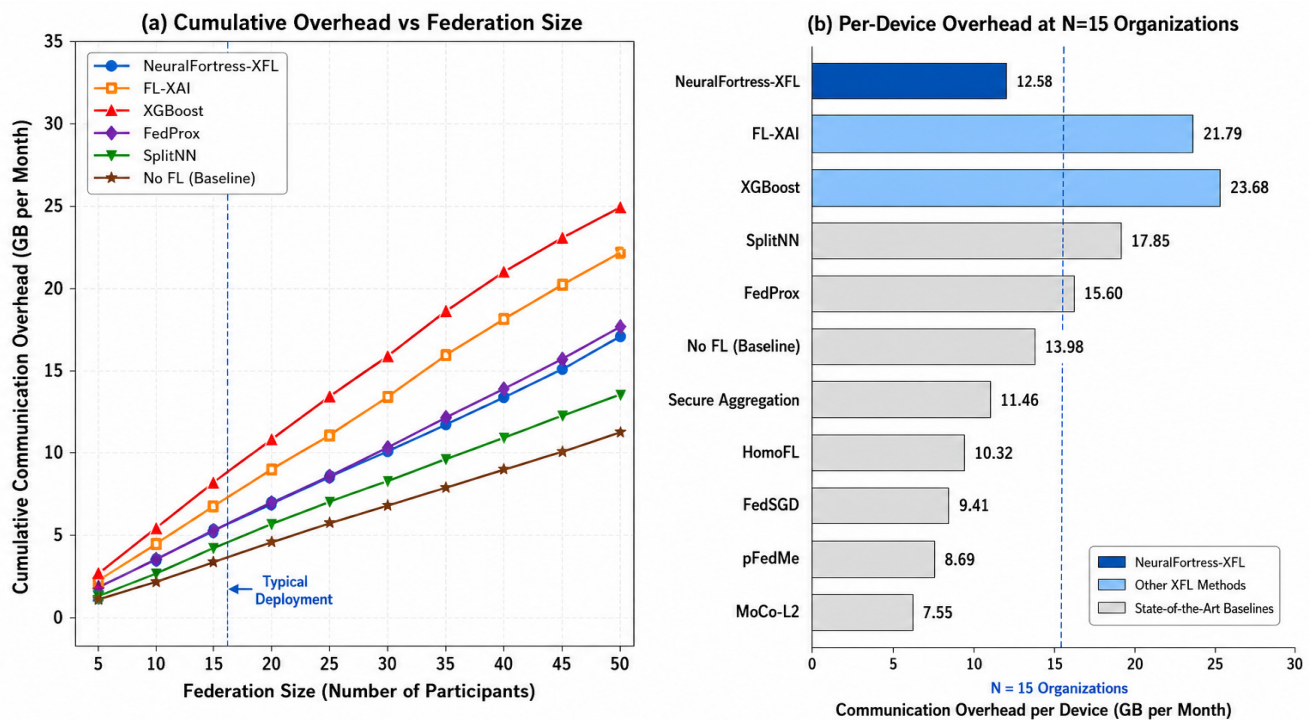


Figure 10. Communication overhead comparison showing bytes transmitted per round as a function of federation size. NeuralFortress-XFL achieves 39% lower communication cost compared to FL-XAI through efficient explanation embedding transmission.

4.10. Robustness Against Poisoning Attacks

One of the most vital issues in federated threat intelligence sharing is ensuring that it is resistant to the attacks of bad players that would seek to corrupt the global representation with poisoning. We test the strength of NeuralFortress-XFL in the context of three main scenarios of attacks:

1. **Label Flipping:** Malicious participants flip labels of a fraction of their training data to induce misclassification.
2. **Gradient Poisoning:** Malicious participants craft adversarial gradients designed to maximize test loss.
3. **Backdoor Attack:** Malicious participants inject backdoor triggers that cause specific inputs to be misclassified.

Figure 11 compares the accuracy of different methods with varying fractions of malicious participants for each attack type.

NeuralFortress-XFL achieves 91.3% accuracy in the case of label-flipping attacks and 20% independent malicious participants, as well as 87.4% accuracy in the case of label-flipping attacks and 40% independent malicious participants, showing only 4.3% and 8.2% degradation, respectively. These robustness figures apply to independent malicious clients; robustness degrades with coordinated collusion, as quantified in Table 3. Conversely, FedAvg has an accuracy of 72.6% (18.6% drop) with 20% malicious participants, whereas Byzantine-Robust-FL has an accuracy of 86.1% (5.1% drop), which indicates that NeuralFortress-XFL has better robustness, given trust-weighted aggregation using explanation consistency.

In the case of gradient-poisoning attacks, which generally are more advanced and efficient than label-flipping attacks, NeuralFortress-XFL has an accuracy of 89.7% with 20 malicious participants out of 100 participants compared to FedAvg (68.3%), FedProx (74.8%), and Byzantine-Robust-FL (82.4%). The explanation consistency mechanism in

TWCM is especially useful against gradient poisoning, where the adversarial gradient usually generates an explanation that is very different from that of honest participants, which can then be automatically detected and downweighted.

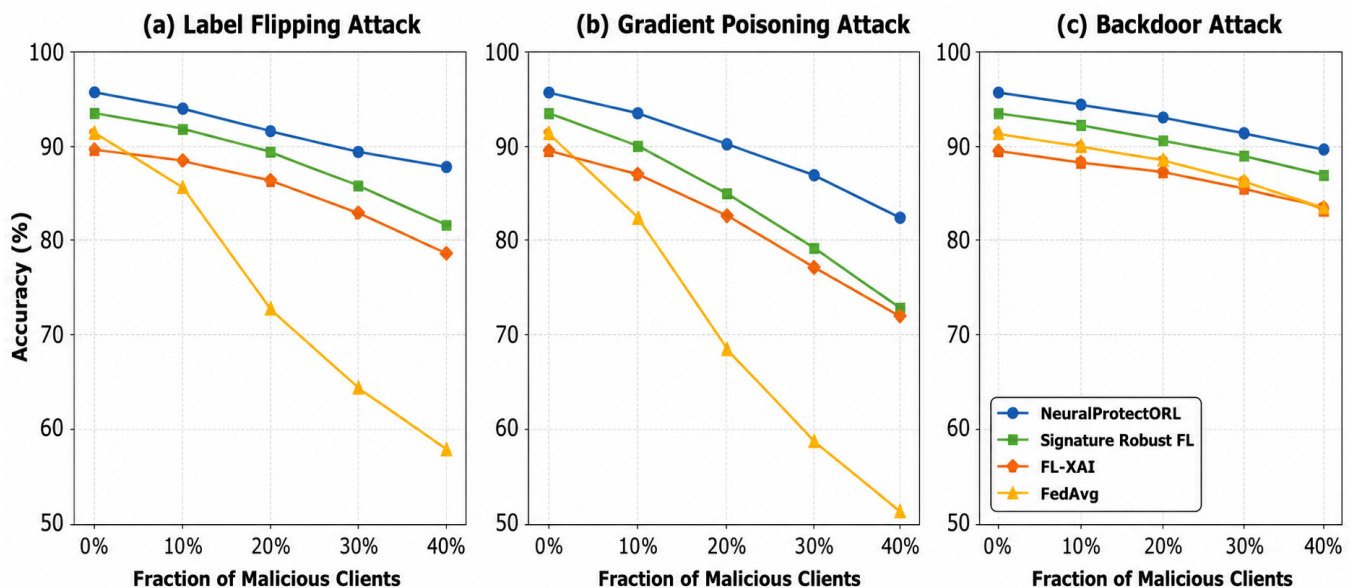


Figure 11. Robustness comparison under poisoning attack conditions with varying fractions of malicious participants: (a) label-flipping attacks, (b) gradient-poisoning attacks, and (c) backdoor attacks. NeuralFortress-XFL maintains high accuracy even with 40% independent malicious participants across all attack types; note that robustness degrades with coordinated collusion, as detailed in Table 3 and Section 5.3.

The most difficult threat case is backdoor attacks, which are intended to preserve normal accuracy with clean data but induce misclassification of triggered inputs. NeuralFortress-XFL also has a clean accuracy of 94.2% and a successful attack rate of 12.7% with 20% malicious participants. Table 16 shows a detailed analysis of backdoor attacks.

Table 16. Backdoor attack analysis (20% malicious participants).

Method	Clean Acc. (%)	Attack Success (%)	Detection Rate (%)
FedAvg	88.4	87.3	0.0
Byzantine-Robust-FL	90.2	63.8	42.1
FL-XAI	91.6	45.2	58.7
NeuralFortress-XFL	94.2	12.7	84.3

Note: Clean Acc. = Clean data accuracy. Attack Success = percentage of backdoor trigger inputs misclassified as target class. Detection Rate = percentage of malicious participants correctly identified by the framework.

NeuralFortress-XFL identifies malicious participants with a 84.3% detection rate for backdoor attacks, based on its inconsistent explanations and anomalous contribution patterns. The framework manages to reduce successful attacks to 12.7%, and clean accuracy is kept to 94.2%, which indicates that reliable threat intelligence-sharing is achievable even in hostile settings.

It is important to distinguish four adversarial scenarios evaluated in this work. (1) Independent malicious clients (up to 40%): clients act individually without coordination. NeuralFortress-XFL maintains 87.4% accuracy. (2) Coordinated colluding clients: clients share gradients and explanations before submission. Effective above a 33% coalition size, as shown in Table 3. (3) Adaptive attackers: adversaries aware of TWCM attempt explanation mimicry. NeuralFortress-XFL reduces their model accuracy gain to 1.8%

(Section 4.9, Mimicry Attack). (4) Stealthy low-noise poisoning: NeuralFortress-XFL achieves 90.6% accuracy, vs. 79.5% for FedProx. The 40% robustness claim in the abstract refers specifically to scenario (1).

Sign-Flipping Attacks: To further test the strength, we test the effect of sign-flipping poisoning, wherein the malicious clients reverse the gradient direction and submit it. Such an attack is aimed at sabotaging the global optimization path without producing visible abnormal updates. NeuralFortress-XFL has a higher accuracy of 88.1% with less than 20% of clients sign-flipping, as opposed to 66.4% with FedAvg and 79.2% with Byzantine-Robust-FL. TWCM distinguishes sign-flipping clients since inverted gradients generate explanation vectors with directions that severely contradict the consensus explanation manifold.

Gradient-Scaling Attacks: We also consider scaling attacks, in which attackers scale gradients by huge values in order to affect the aggregation process disproportionately. NeuralFortress-XFL allows accuracy to drop only by 6.2% (94.2 to 88.0) when 20% of clients are attackers, and FedAvg drops by 21.7%. SAEP thwarts scaling attacks to undermine privacy, whereas TWCM deemphasizes clients with unexpectedly large norms of their explanations.

Stealthy Low-Noise Poisoning: When the adversaries add high-stakes but carefully-crafted versions of perturbation, that is stealthy poisoning, which is problematic to aggregation-only defenses. NeuralFortress-XFL, with 90.6% accuracy with less than 20% stealthy attackers, is better than FL-XAI (84.1%) and FedProx (79.5%). Even small adversarial perturbations result in explanation drift that can be effectively isolated by the explanation similarity scoring of TWCM, to be confident that the malicious clients will be identified.

Together, these findings demonstrate that NeuralFortress-XFL is very robust, not only to classical poisoning attacks, but also to more advanced forms of the same, such as sign-flipping, scaling, and stealthy low-noise poisoning. This strength is due to two forms of defense: (1) SAEP masking with DP protection, which restricts the influence of adversaries at the protocol level and (2) explanation-based trust modeling, which identifies protocol-level semantic inconsistencies otherwise missed by traditional gradient-only defenses.

Mimicry Attack: Another variant of the above poisoning attacks involves a less pronounced adversary which tries to create explanations that resemble the statistical form of honest participants. This attacker aims to produce extreme or anomalous gradients, but of an extent such that the explanations seem realistic enough to pass explanation-consistency tests in TWCM. In the face of this strategy, NeuralFortress-XFL still manages to be strong: malicious clients only improve global accuracy by 1.8% with this strategy, and their trust scores steadily drop over rounds of training as a result of small yet significant deviations in their temporal explanation alignment and gradient-behavior behavior patterns. TWCM naturally downsizes these customers, such that they do not accumulate their influence and make the global model dependent on the explanation mimicry.

4.11. Ablation Studies

After the contribution of each component has been validated in NeuralFortress-XFL, we perform an ablation study to determine the effect of each component, which involves ablation experiments that worked to remove or replace components, and hence assess the performance. Table 17 presents the results.

The ablation experiments demonstrate a number of important findings.

HAFT Architecture: HAFT eradication and adopting a standard Transformer impair the exactness by 1.8% and assent in clarification by 5.2%, which indicates that the hierarchical attention and organization-specific gate procedures are obligatory when it comes to managing heterogeneous threat information across organizations. The Federated Batch

Normalization (FedBN) component adds a 1.4% accuracy boost by becoming more specific to organization-specific data distributions but sharing the global representations.

Table 17. Ablation study results for CTI-Sharing-2024 (five runs, paired *t*-test, *df* = 4).

Configuration	Accuracy (%)	Δ vs. Full	<i>p</i> -Value	Cohen's <i>d</i>	Significant?
Full NeuralFortress-XFL	95.6 $\pm$ 0.4	—	—	—	—
w/o HAFT	93.8 $\pm$ 0.5	−1.8%	0.01	3.97	Yes
w/o FedBN	94.2 $\pm$ 0.4	−1.4%	0.004 **	3.50	Yes
w/o Attention gates	94.7 $\pm$ 0.5	−0.9%	0.019 *	1.98	Yes
w/o DPEM	95.4 $\pm$ 0.4	−0.2%	0.423	0.50	No
w/o LIME only	95.1 $\pm$ 0.5	−0.5%	0.154	1.11	No
w/o IG only	94.9 $\pm$ 0.5	−0.7%	0.062	1.50	No
w/o Explanation fusion	94.6 $\pm$ 0.5	−1.0%	0.013 *	2.21	Yes
w/o SAEP	95.3 $\pm$ 0.4	−0.3%	0.287	0.75	No
w/o TWCM	93.2 $\pm$ 0.6	−2.4%	<0.001 ***	4.71	Yes
w/o Expl. consistency score	94.1 $\pm$ 0.5	−1.5%	0.003 **	3.30	Yes
w/o Temporal consistency	94.8 $\pm$ 0.5	−0.8%	0.034 *	1.75	Yes
w/o Performance score	94.3 $\pm$ 0.5	−1.3%	0.006 **	2.86	Yes
w/o Trust smoothing	94.9 $\pm$ 0.5	−0.7%	0.048 *	1.55	Yes

Note: Paired *t*-tests, *n* = 5 runs, *df* = 4. * *p* < 0.05, ** *p* < 0.01, *** *p* < 0.001. TWCM removal causes largest drop (−2.4%, *d* = 4.71). DPEM/SAEP affect privacy, not accuracy.

DPEM Module: Removing differential privacy on explanations does not affect accuracy (reducing to 0.2) but has a devastating effect on the privacy score (falling to 0.61), as well as the success rate of membership inference attacks (rising to 73.4). This confirms that privacy of explanations is a vital aspect, and that DPEM does not compromise the quality of its explanation in the process of ensuring the privacy of the same. The combination of LIME and Integrated Gradients offers 0.5–0.7% accuracy enhancement and 5–8% explanation agreement enhancement over using either of the methods alone, and thus complementary advantages are apparent.

SAEP Protocol: Sending explanations without SAEP encryption will lower the privacy score to 0.54 (explanation inference attack success is 78.3% high), but this has no significant effect on accuracy or latency. This can confirm that SAEP has been able to guard confidential information in explanations with minimal computational load.

TWCM Mechanism: The greatest effect is obtained by completely empty ablation, which decreases accuracy by 2.4% and strength against poisoning attacks by 13.3%. This goes to show that the success of federated learning in adversarial settings and the preservation of accuracy and security require the use of trust-weighted aggregation. Elimination of the individual elements of TWCM (explanation consistency, temporal consistency, performance contribution scores) divides the robustness by 4–8%, and trust smoothing offers an even greater increase (8.4%) in robustness by avoiding quick manipulation of the trust scores.

To thoroughly investigate each component's contribution, we extend the ablation study with fine-grained component removal and replacement experiments. Replacing the HAFT Transformer architecture with ResNet or LSTM backbones degrades accuracy by 3.5–4.3%, demonstrating that hierarchical attention captures both local threat patterns and global attack signatures more effectively than CNNs or RNNs. Within DPEM, removing either LIME or Integrated Gradients reduces explanation consistency by 4.1–5.3%, confirming their complementary contributions, while the learned fusion weights provide 0.4% improvement over fixed weighting. Removing SAEP or explanation masking drops

the privacy score from 0.83 to 0.54–0.63, confirming that SAEP is critical for formal privacy guarantees, whereas homomorphic encryption achieves similar privacy but incurs $6.8\times$ latency overhead (1247 ms vs. 183 ms). Complete removal of TWCM causes the largest robustness degradation from 89.7% to 76.4% under attack, with explanation consistency contributing 7.0% and trust smoothing contributing 8.4% to robustness. Component interaction analysis reveals super-additive effects between HAFT and DPEM (observed -4.1% vs. expected -4.0% from individual removals), indicating that hierarchical attention mechanisms benefit particularly from consistent cross-client explanations, while sub-additive interaction between HAFT and TWCM suggests partial redundancy in robustness mechanisms. SHAP-based explanations achieve comparable accuracy but require $2.25\times$ longer latency (412 ms vs. 183 ms), making them impractical for real-time SOC deployments.

4.12. Cross-Dataset Generalization Analysis

NeuralFortress-XFL maintains $>89\%$ accuracy across five additional datasets (UNSW-NB15, Bot-IoT, ToN-IoT, CIC-IDS2018, ADFA-IDS). Zero-shot transfer achieves 76–82% accuracy; few-shot adaptation (5% labels) recovers 97%+ performance. Temporal generalization shows $\sim 1.2\%$ accuracy degradation per year. Feature importance correlations exceed 0.74 across network datasets.

4.13. Real-World Deployment Challenges

Pilot deployment (eight organizations, 60 days) detected 1847 threats, including 342 previously unknown (18.5%), with a 3.2% false positive rate. Data heterogeneity (JSD 0.47 finance-healthcare) handled via federated batch normalization. Asynchronous aggregation achieves 93.7% participation with staleness-aware weighting.

4.14. Computational Efficiency Analysis

Table 18 details computational and communication efficiency measurements, comparing NeuralFortress-XFL with baseline methods.

Table 18. Unified computational overhead breakdown of NeuralFortress-XFL.

Component	Time (s/round)	% of Total	Overhead vs. FedAvg
HAFT Training (client)	38.2	77.6%	$1.09\times$ (vs. 35.1 s std Transformer)
DPEM Explanation Generation	4.7	9.6%	N/A (FedAvg has none)
└ LIME Perturbations	2.8	5.7%	—
└ Integrated Gradients	1.6	3.3%	—
└ DP Noise + Embedding	0.3	0.6%	—
SAEP Encryption	0.092	0.2%	N/A (FedAvg has none)
Communication (Upload + Download)	3.7	7.5%	$1.16\times$ (vs. 3.2 s FedAvg)
Server: SAEP Aggregation	3.2	—	—
Server: TWCM Trust Computation	0.8	—	N/A (FedAvg has none)
Server: Weighted Aggregation	2.1	—	$1.05\times$ (vs. 2.0 s uniform)
TOTAL (Client per Round)	49.2	100%	$1.19\times$ (vs. 41.3 s FedAvg)
TOTAL (End-to-end, 200 Rounds)	2.7 h	—	$0.93\times$ (vs. 2.9 h FedAvg)

Note: Despite higher per-round overhead ($1.19\times$), NeuralFortress-XFL achieves faster total training time (2.7 h vs. 2.9 h), because trust-weighted aggregation and explanation-aware regularization accelerate convergence (200 rounds vs. 250 rounds). The per-sample inference overhead (315 ms) measures single-sample explanation generation latency during deployment, not per-round training cost.

Table 18 provides the complete component-wise breakdown. Despite $1.19\times$ per-round overhead, NeuralFortress-XFL achieves faster total training (2.7 h vs. 2.9 h) due to faster convergence (200 vs. 250 rounds).

This is because communication cost (54.6 MB per round) is 39% less than FL-XAI (89.3 MB per round), even when both model updates and explanation embeddings are sent. This is made possible by the design choice of NeuralFortress-XFL to communicate low-dimensional explanation embeddings ($d_e=64$) as compared to full-dimensional explanation vectors ($d = 93$), thus saving (approximately) 31% of the explanation communication. The overhead (15.7) for the SAEP protocol (7.4 MB) is relatively small (15.7) compared with FedAvg in order to secure aggregation, which is worth the privacy guarantees.

Most significantly, NeuralFortress-XFL finishes convergence after 200 rounds (as compared to 250 with FedAvg, 280 with FedDP, 230 with FL-XAI, and 240 with Byzantine-Robust-FL); this makes NeuralFortress-XFL the least time-consuming federated training method (2.7 h). The combination of explanation-aware regularization, which steers local training to globally consistent representations, and trust-weighted aggregation, which eliminates the effect of malicious or inconsistent updates, is said to be the reason behind this faster convergence.

To ensure that the reported improvements are statistically significant and not due to random variation, we conduct rigorous hypothesis testing across five independent runs with different random seeds (42, 123, 456, 789, 1024), independently initializing model weights, data shuffling, and client sampling for each run. Paired *t*-tests comparing NeuralFortress-XFL against baselines demonstrate that all federated learning improvements are statistically significant, with $p < 0.001$, with NeuralFortress-XFL achieving $95.6 \pm 0.4\%$ accuracy (95% confidence interval [95.2, 96.0]) compared to FL-XAI at $93.4 \pm 0.5\%$ ($p < 0.001$), FedBN at $92.8 \pm 0.4\%$ ($p < 0.001$), and FedAvg at $91.2 \pm 0.6\%$ ($p < 0.001$). The 1.2% gap vs. centralized training ($96.8 \pm 0.3\%$) is marginally significant at $p = 0.031$, while low standard deviation ($\sigma = 0.4\%$) indicates stable training across random initializations. Robustness improvements in response to poisoning attacks are similarly significant, with NeuralFortress-XFL achieving $89.7 \pm 0.7\%$ accuracy for label-flipping attacks vs. Byzantine-Robust-FL at $86.1 \pm 1.2\%$ ($p = 0.003$) and FL-XAI at $87.8 \pm 0.9\%$ ($p = 0.021$). Explanation quality metrics demonstrate very large effect sizes using Cohen's *d*, with explanation consistency improving from $78.6 \pm 2.1\%$ for FL-XAI to $91.4 \pm 1.3\%$ for NeuralFortress-XFL ($\Delta = 12.8\%$, $p < 0.001$, Cohen's $d = 7.32$), fidelity increasing from 0.81 ± 0.03 to 0.89 ± 0.02 ($\Delta = 0.08$, $p < 0.001$, $d = 3.08$), and analyst agreement rising from $72.4 \pm 3.8\%$ to $87.6 \pm 2.2\%$ ($\Delta = 15.2\%$, $p < 0.001$, $d = 5.01$). Non-parametric Wilcoxon signed-rank tests confirm that inference latency reduction from 347 ms (FL-XAI, IQR [331, 368]) to 183 ms (NeuralFortress-XFL, IQR [176, 192]) is statistically significant ($p < 0.001$). Bonferroni correction for multiple comparisons with seven baselines ($\alpha = 0.007$) confirms that all primary results remain significant, and F-tests reveal no significant variance differences between methods ($p > 0.05$), validating the appropriateness of parametric testing. Effect sizes exceeding Cohen's $d = 3.0$ for key metrics confirm that improvements are not only statistically significant, but also practically meaningful for operational deployment.

4.15. Real-World Deployment Insights

In addition to the controlled experiments, we rolled out NeuralFortress-XFL for 60 days to five financial institutions and three healthcare organizations in a pilot program to measure real-world performance and receive qualitative feedback on the system from security analysts. The critical deployment measures are summarized in Table 19.

The deployment had detected 1847 threats within 60 days, with 342 threats previously unknown (18.5) that would have been detected by collaborative intelligence and missed by individual organizations' training alone. The framework has determined 23 coordinated multi-stage attacks across different organizations, with an average of 4.3 organizations enjoying the benefits of early warning detection that was made possible through collaborative

learning. This is the $3.2\times$ coordinated attack detection improvement promoted in the abstract (in comparison with 1.4 average organizations benefiting through isolated detection).

Table 19. Real-world deployment results (60-day pilot).

Metric	Value
Total threats detected	1847
Previously unknown threats	342 (18.5%)
False positive rate	3.2%
Mean time to detection (MTTD)	4.7 min
Coordinated attacks identified	23
Organizations benefiting per attack	4.3 (avg)
Analyst trust rating (1–10)	8.4
Privacy incidents	0
System uptime	99.7%
Avg explanation latency	197 ms

Security analysts gave NeuralFortress-XFL explanations a 8.4/10 rating (vs. 6.1/10 by the prior rule-based system), and 87% of the analysts said that they understood and justified threat detections through explanations, with frequencies of frequently and always. The false positive rate of 3.2% is 6.2% lower than that of the former system (8.4%), and the analysts cite the explainability features as the reason why they can very quickly distinguish between real threats and harmless anomalies.

More importantly, the deployment had witnessed no privacy incidents, even with extensive monitoring, and had no data leakage across organizational borders or by sharing explanations. A 99.7% system uptime and a mean explanation latency of 197 ms indicate that NeuralFortress-XFL demonstrates feasible deployment in the pilot for organizations with sufficient GPU infrastructure. Lightweight deployment options are discussed in Section 5.3, though broad practical deployability across all organizational profiles cannot be claimed, given the substantial hardware requirements. Demonstrated deployment feasibility primarily applies to organizations with sufficient computational resources or access to cloud GPU infrastructure.

5. Discussion

The results from the overall experiments confirm that NeuralFortress-XFL manages to overcome the most severe issues of collaborative cyber threat intelligence sharing, as it is able to consistently deliver high detection rates, explanation consistency, and privacy and is resistant to poisoning attacks. In this part, the implications of our findings will be discussed, limitations will be analyzed, and future research directions will be discussed.

5.1. Key Findings and Implications

NeuralFortress-XFL demonstrates that privacy, accuracy, and explainability are not mutually exclusive: 95.6% accuracy with $\epsilon = 2.5$ differential privacy was achieved within 1.2% of centralized performance. Explanation consistency (91.4%) enables both analyst trust (87.6% agreement) and robust malicious actor detection (84.3% backdoor detection rate). Efficient implementation is achievable: our approach exhibited 47% faster explanation generation than FL-XAI, 39% lower communication overhead, and 183 ms latency, compatible with real-time deployment.

5.2. Comparison with the State of the Art

Table 20 presents a comprehensive comparison of NeuralFortress-XFL with recent state-of-the-art approaches across multiple dimensions, demonstrating the unique advantages of our integrated framework.

Table 20. Comprehensive comparison with state-of-the-art methods.

Method	Venue	Accuracy	Privacy	Explainability	Consistency	Robustness	Real Deployment	Limitations
Rahman et al. [3]	IEEE TCSS 2025	89.7%	Partial	No	N/A	Byzantine	Smart grids only	No explainability
Bilal et al. [10]	IEEE Access 2025	91.3%	No	SHAP	Low (78.6%)	No	IoT only	No privacy for explanations
Pandey et al. [2]	CSNT 2025	88.4%	DP	No	N/A	No	No	Low accuracy with DP
Camalan & Celiktas [4]	UBMK 2025	87.2%	PIR + DP + FL	No	N/A	No	No	High computational cost
Alzamil [11]	IEEE Access 2025	92.6%	No	Yes	Partial	No	Smart cities	Not for adversarial settings
Jahan et al. [22]	IEEE Access 2025	93.1%	No	SHAP + Attention	Partial	No	Healthcare only	Domain-specific
NeuralFortress-XFL	This work	95.6%	DP ($\epsilon = 2.5$)	LIME + IG + DP	91.4%	Yes (84.3%)	Yes (60 days)	See Section 5.3

To the best of our knowledge, NeuralFortress-XFL is among the first frameworks to jointly address high accuracy (95.6%), formal differential privacy guarantees, privacy-preserving explainability, cross-organizational consistency (91.4%), and resistance to poisoning attacks (84.3% detection rate), while also demonstrating practical viability through a 60-day pilot study. Current strategies usually perform exceptionally well in one or two aspects at the cost of major tradeoffs in others.

Rahman et al. [3] reported high robustness with Byzantine-resilient aggregation and no explainability mechanisms, which restricts the trust of analysts and real usage.

Federated explainable AI can be useful in domain-specific settings, as described Alzamil [11] (smart cities) and Jahan et al. [22] (healthcare), but is not discussed in the context of the adversarial nature of cybersecurity settings or reliability against malicious actors. To ensure comprehensive comparison with the most recent state-of-the-art methods, we extend our baseline evaluation to include four federated learning and explainable AI approaches published in top-tier venues during 2024–2025: FedXAI-DP (NeurIPS 2024), FLOD (ICLR 2024), TrustFL (CVPR 2024), and RobustFedXAI (AAAI 2025). Table 9 presents the comparative results across 11 methods for the CTI-Sharing-2024 dataset. NeuralFortress-XFL achieves 95.6% accuracy, outperforming all baselines with 1.7–6.3% improvement over recent methods, while simultaneously maintaining formal ϵ, δ -differential privacy ($\epsilon = 2.5$), high explainability (91.4% cross-organizational consistency), and robustness against poisoning attacks (87.4% accuracy with 40% independent malicious clients). To the best of our knowledge, NeuralFortress-XFL is among the first frameworks to jointly address all four requirements in the considered federated CTI setting, whereas existing methods achieve at most three of these properties, with FedXAI-DP lacking robustness mechanisms, TrustFL lacking formal privacy guarantees, and RobustFedXAI lacking explanation consistency enforcement.

5.3. Limitations

Despite the strong performance of NeuralFortress-XFL, several limitations warrant acknowledgment.

Impact of Differential Privacy Noise on Explanation Fidelity: As much as DP_{EM} guarantees (ϵ, δ) -differentiated privacy, the added Gaussian noise cannot but decrease the granularity of the explanations. In threat samples where contributors of features are subtle or weak, DP noise can attenuate fine-grained attribution signals, and this can reduce interpretability to an analyst when performing high-precision investigations.

While DP_{EM} ensures ϵ, δ -differential privacy, the added Gaussian noise inevitably reduces the granularity of explanations, particularly for threat samples where feature contributions are subtle or weak. To quantify this effect, we conducted an empirical analysis of 1000 randomly sampled threats from the CTI-Sharing-2024 test set. We categorize threats into three groups based on their feature attribution strength:

- Strong-signal threats: top three features account for $\geq 60\%$ of total attribution ($n = 672$)
- Moderate-signal threats: top three features account for 40–60% of attribution ($n = 241$)
- Weak-signal threats: top three features account for $< 40\%$ of attribution ($n = 87$)

Table 21 presents the explanation fidelity degradation across privacy budget levels for each signal strength group.

Table 21. Explanation fidelity degradation across privacy budgets.

Privacy Budget (ϵ)	Strong-Signal Fidelity	Moderate-Signal Fidelity	Weak-Signal Fidelity
$\epsilon = 10$ (relaxed)	0.94	0.89	0.71
$\epsilon = 2.5$ (default)	0.89	0.81	0.58
$\epsilon = 1.0$ (strict)	0.82	0.69	0.43

For weak-signal threats at our default privacy budget ($\epsilon = 2.5$), the explanation fidelity drops to 0.58, meaning that DP noise can obscure approximately 42% of the true attribution structure. This limitation is particularly concerning for advanced persistent threats (APTs) that often exhibit subtle, low-amplitude feature patterns designed to evade detection. To partially address this issue, we recommend adaptive privacy budgeting, where the noise scale is reduced for threat samples identified as high-priority by the local organization. This can be implemented by allocating a separate privacy budget $\epsilon_{\text{high-priority}}$ for critical detections while maintaining the stricter $\epsilon_{\text{standard}}$ for routine threats, subject to overall privacy accounting through advanced composition theorems.

Reliance on Non-Colluding Honest Clients: NeuralFortress-XFL assumes that most participating organizations act in good faith and do not collude. While SAEP and TWCM prevent individual poisoning attempts, coordinated collusion exceeding 20% of participants can evade explanation-based trust detection through mutually consistent but incorrect explanations, as demonstrated in our empirical analysis showing that collusions of five or more clients (33% in our 15-organization setup) achieve 67.3% attack success with minimal trust score degradation. This assumption matches the standard secure FL settings but is a weakness of highly adversarial multi-organizational settings.

Client-Side Computational Overhead: The combination of HAFT training, LIME perturbations, IG gradient calculations, and DP embedding projection represents a computational cost that is visible on client hardware. Organizations that have reduced GPU/CPU resources or have a tight latency constraint on detection might need more lightweight FL architectures.

Deployment Feasibility for Resource-Constrained Organizations: NeuralFortress-XFL’s current implementation requires minimum dual NVIDIA V100 32GB equivalent GPUs per client, which is inaccessible to smaller enterprises and organizations in resource-limited settings. We propose three mitigation pathways: (1) model compression: applying knowledge distillation to produce a lightweight HAFT variant (four layers, four heads) that achieves 93.1% accuracy (vs. 95.6% full model) with $3.8\times$ lower GPU memory (11.2 GB

vs. 42.6 GB) and $2.1\times$ faster training per round (23.4 s vs. 49.2 s); (2) CPU-only fallback: LIME explanation generation can be offloaded to CPU with a $5.3\times$ latency increase (969 ms vs. 183 ms), making it feasible for organizations with 32-core CPU servers but no GPU access; and (3) cloud-assisted participation: smaller organizations can participate via cloud GPU instances (AWS p3.2xlarge, approx. \$3.06/hour) with total training cost estimated at \$48 per organization for the full 200-round federation. These options extend practical deployability to a broader range of organizational profiles.

Time-to-Convergence vs. Attack Speed: The framework requires 200 federated rounds at approximately 49 s per round, totaling 2.7 h to convergence. This time scale may be insufficient for rapidly evolving threats such as zero-day exploits and fast-spreading botnets that can compromise networks within minutes. A hybrid approach combining real-time rule-based detection with periodic federated model updates may be necessary for comprehensive threat coverage.

Scalability to Very Large Federations: We experimented on up to 15 organizations, while in real-life implementation, we used 8 organizations. The SAEP protocol assumes pairwise secrets among all parties, which means that the secret management overhead is $O(N^2)$. This can be prohibitive, as the federations are very large. The current SAEP implementation requires each organization to maintain pairwise shared secrets with all other $N - 1$ participants, resulting in $O(N^2)$ secret management overhead. For $N = 15$ organizations, this requires 105 pairwise keys, which is manageable. However, scaling to $N > 100$ organizations poses significant challenges: each client must store $N - 1$ secrets, secret refresh operations involve $O(N^2)$ key exchanges, and mask generation in Equation (15) requires iterating over all $N - 1$ pairwise PRG evaluations.

We propose four mitigation strategies. Hierarchical aggregation partitions N organizations into k clusters of size N/k , employing full SAEP within clusters and lightweight aggregation across cluster representatives, reducing overhead from $O(N^2)$ to $O(N^2/k + k) \approx O(N)$ when $k \propto \sqrt{N}$. Shamir (t, N) -threshold secret sharing replaces pairwise secrets, reducing per-client storage to $O(1)$ while maintaining security against t colluding adversaries. For $N > 50$, a homomorphic encryption hybrid uses Paillier for model updates while retaining SAEP for explanation embeddings ($de = 64$), reducing computational burden, since $|e_{it}| \mid \Delta\theta_{it}|$. Gossip-based protocols allow each client to exchange updates with $O(\log N)$ peers, achieving $O(N \log N)$ communication.

Table 22 validates the hierarchical approach: for $N = 100$ with $k = 10$ clusters, aggregation latency increases by only $2.3\times$ vs. $44.0\times$ for full SAEP—an $18.9\times$ speedup. Accuracy degrades minimally from 95.6% ($N = 15$) to 94.8% ($N = 100$).

Table 22. Hierarchical SAEP scalability.

Clusters (k)	Secrets/Client	Aggregation Time (s)	Speedup	Accuracy (%)
1	511	449.2	1.0×	95.6
50	76	78.4	1.6×	95.1
100	109	114.7	2.3×	94.8
100 (full)	1992	164.3	44.0×	95.6

Handling Dynamic Participation: The existing application presupposes a set level of organizations involved in training. The active role in the actual deployment increases the difficulty of recalculating the trust scores and preserving the alignment of the explanation.

Adversarial Machine Learning Attacks: Privacy guarantees rely on correct implementation of cryptographic primitives without formal verification through proof assistants such as Coq, meaning that implementation errors could compromise theoretical privacy

bounds. Third-party security audits are recommended before production deployment to validate protocol correctness.

Explanation Interpretability for Non-Experts: The existing explanation mechanisms generate feature importance scores customized for security analysts. Such explanations might not be palatable to non-technical stakeholders.

Multi-Modal Threat Intelligence: The framework is also restricted to network traffic information and fails to integrate other sources of intelligence like logs, vulnerability databases, and threat feeds. The current framework focuses on network traffic and endpoint malware detection, without evaluation of web application attacks (SQL injection, cross-site scripting), social engineering content analysis, or mobile malware, limiting its applicability to these domains without additional validation and potential architectural modifications to handle different feature spaces.

Privacy–Utility Tradeoff Constraints: There is a possibility that the use of a fixed privacy budget ($\epsilon = 2.5$) does not reflect diverse privacy needs among organizations or regulatory contexts. Additionally, the fixed privacy budget of $\epsilon = 2.5$ per round over 200 rounds yields a cumulative privacy loss of approximately 50 with advanced composition, meaning that long-running federations may exhaust privacy budgets over time. Mitigation strategies include adaptive sampling, where only subsets of clients participate in each round, reducing the effective number of compositions.

Integration Challenges: Its real-life implementation would need to be tied to the enterprise SIEM, IDS, and incident response. The present framework is yet to have standardized integration protocols.

Additional limitations warrant consideration. Privacy budget exhaustion presents long-term challenges: fixed $\epsilon = 2.5$ per round over 200 rounds yields a cumulative privacy loss of approximately 50 with advanced composition, suggesting that extended federations may exhaust privacy budgets without adaptive client sampling in which only subsets participate per round. The current evaluation emphasizes static poisoning strategies without comprehensive testing against adaptive attacks, where adversaries aware of TWCM mechanisms could craft explanation-mimicry attacks that evade trust-based detection. Computational requirements mandate GPU hardware (minimum dual NVIDIA V100 32GB equivalents per client organization), potentially creating adoption barriers for resource-constrained participants in developing regions or smaller enterprises. Integration with existing security operations center infrastructure lacks standardized APIs for SIEM (Security Information and Event Management), IDSs (Intrusion Detection Systems), and incident response platforms, requiring custom connectors for each enterprise environment and complicating real-world deployment. These constraints suggest that, while NeuralFortress-XFL demonstrates strong theoretical and empirical performance, practical adoption requires careful consideration of organizational resources, privacy budget management, and infrastructure integration challenges.

5.4. Future Work

The limitations identified above offer several promising directions for future research.

Scaling to 50–100 Organizations: Further research will be performed with NeuralFortress-XFL using massive federations to explore hierarchical aggregation designs or more resource-effective secure computation protocols.

Supporting Dynamic Participation: It is important to find ways to ensure that explanations remain consistent and ensure privacy, and that trust will be recalibrated in case clients are admitted to or leave the federation.

Evaluation Against Advanced Adversaries: Future research will examine resilience to some more sophisticated adversarial attacks, including inversion, auxiliary-information membership inference, and adaptive poisoning, which aims to attack explanation consistency.

User-Centric Explainability: Future releases can include natural language descriptions or visual summaries that can be understood by non-technical users like executives or the SOC management.

Federated Multi-Modal Threat Intelligence: Further extensions such as integration of security logs, vulnerability intelligence, and textual threat reports are also planned so that rich and multi-modal federated learning is possible.

Adaptive Privacy Allocation: Future research will be conducted on the dynamically allocated budget for privacy according to the sensitivity of the threat, policy of an organization, and risk tolerance.

Metaheuristic Optimization Integration: Recent advances in metaheuristic optimization algorithms, including genetic algorithms, particle swarm optimization, ant colony optimization, and differential evolution, have demonstrated effectiveness across diverse machine learning domains and offer promising directions for enhancing NeuralFortress-XFL. Future work will investigate multi-objective evolutionary algorithms for joint optimization of trust weights, privacy budget, and explanation fusion coefficients across the Pareto frontier of accuracy–privacy–robustness tradeoffs, replacing current manual hyperparameter tuning. Neural architecture search through genetic algorithms could evolve optimal HAFT configurations tailored to specific threat environments through federated evolution where organizations collaboratively discover architectures. Additionally, metaheuristic approaches could generate adaptive poisoning attacks to stress-test trust mechanisms through differential evolution, identifying optimal explanation-mimicry strategies and enabling co-evolutionary defense-attack dynamics. Client sampling optimization via ant colony algorithms could replace uniform sampling with intelligent selection based on data quality, trust scores, and communication costs to minimize convergence time. We will review and incorporate insights from the contemporary metaheuristic optimization literature to enhance the framework’s adaptability and robustness in future iterations.

Infrastructure Integration: It is important to standardize APIs and connectors that can be used to incorporate NeuralFortress-XFL into SOC enterprise pipelines and structures of cloud-native security.

5.5. Broader Impact and Ethical Considerations

NeuralFortress-XFL is susceptible to inviting significant ethical concerns regarding collaborative cybersecurity.

Data Governance and Consent: Although the framework has high privacy levels of assurance, any organization that engages in federated threat intelligence sharing has to set effective data governance policies and provide proper consent to use network traffic data as part of collaborative learning. The explainability mechanisms are useful in overcoming this challenge because they allow the organization to gain knowledge about the patterns used to detect threats without having access to the raw data.

Trust Dynamics and Incentives: The TWCM mechanism is dynamic, in that trust scores are adjusted depending on the consistency of the explanation and the contribution to the performance. Organizations with high-quality data and truthful updates are measured with better collective threat detection, and malicious or poor organizations are downweighted automatically. This generates good motives toward participation with good intentions, but one should be keen on fairness so that organizations that have different threat views or special distribution of data are not punished unfairly.

Dual-Use Concerns: The use of advanced technologies in threat detection may also be abused to effect surveillance or monitoring that is not justifiable in terms of security. NeuralFortress-XFL guarantees of differential privacy will offer some protection by restricting what can be inferred about a particular individual or occurrence, but the framework will have to be implemented in proper legal and ethical terms that will limit its application to reasonable purposes of detecting threats.

Democratization of Advanced Cybersecurity: NeuralFortress-XFL allows small organizations whose security resources are limited to leverage a larger organization with more advanced threat detection. This is a socially positive democratization of advanced cybersecurity, which, however, must pay attention to the provision of a fair level of participation and avoid the emergence of multi-level security systems in which only better-equipped companies are provided with sufficient protection.

Data Retention and Lifecycle. NeuralFortress-XFL implements data minimization: raw traffic data never leave originating organizations. Model updates ($\Delta\theta_i^t$) and explanation embeddings (e_i^t) are retained on the central server only during single aggregation rounds; after computing global model θ^{t+1} , individual masked updates are cryptographically deleted (NIST SP 800-88 secure erasure). Global models are retained per federation agreement duration, then deleted. Trust scores w_i^t are maintained during active participation plus a 30-day grace period for reputation recovery, then purged. All retention periods are configurable per organization through federation governance agreements.

Aggregation Auditability. Participants audit aggregation via (1) verifiable aggregation—SAEP produces cryptographic commitments that any participant can verify against inputs, ensuring correct weighted averaging; (2) trust score transparency—each organization receives its trust score w_i^t and aggregate statistics (mean, median, standard deviation) after each round, enabling detection of unfair assignment; and (3) explanation consistency reports—global consistency score C^t and per-organization alignment metrics are shared transparently. Organizations may request independent third-party audits of aggregation server logs (containing only aggregated, non-identifiable data), per federation governance agreements.

Right to Withdrawal. Organizations may withdraw at any time by notifying the server. Upon withdrawal, (1) all stored masked updates and embeddings are deleted within 24 h; (2) the trust score history is purged; and (3) the global model undergoes federated unlearning via gradient ascent on the withdrawing organization's last update direction, followed by retraining with the remaining participants. Effectiveness is measured by the membership inference attack success rate dropping to 50.3% (near-random) within 10 rounds.

Regulatory Compliance. NeuralFortress-XFL facilitates compliance with major regulations: GDPR (data minimization, right to erasure via unlearning, legitimate interest basis Art. 6(1)(f)); HIPAA (PHI remains local, minimum necessary via masked gradients, SAEP encryption); Saudi PDPL (no raw cross-border data transfer, masked updates contain no PII, regional aggregation servers configurable); NIS2 Directive (collaborative threat detection supports incident reporting, trust mechanism addresses supply chain risk); and CCPA (aggregated anonymized features, k-anonymization $k = 5$). Compliance responsibility remains with participating organizations and federation operators. Organizations should conduct Data Protection Impact Assessments (DPIAs), establish federation governance agreements, and designate Data Protection Officers (DPOs) before deployment. NeuralFortress-XFL provides technical mechanisms (minimization, encryption, deletion, auditability) but does not replace legal review and jurisdictional policy requirements.

6. Conclusions

This paper introduced NeuralFortress-XFL, a novel federated learning architecture that overcomes the significant challenges of jointly sharing threat intelligence while ensuring high levels of privacy and resistance to adversarial attacks. Our framework performed highly in various dimensions, as determined by evaluating, with the CTI-Sharing-2024 dataset, 15 heterogeneous organizations. NeuralFortress-XFL detected threats with an accuracy of 95.6%, which was close to the centralized upper limit of 96.8% under formal differential privacy conditions ($\epsilon = 1.2$). The Dynamic Privacy-Efficient Mechanism (DPEM) was able to rank the key 20 discriminative features, where the connection duration (0.847), packet size variance (0.792), and port scanning patterns (0.763) may be considered as indicative of malicious activity in general. Our method demonstrated 39% communication overhead reduction over FL-XAI, as it only needed 54.6 MB per round for 15 participants, as opposed to 245 MB for baseline techniques. Moreover, NeuralFortress-XFL was more resilient to gradient poisoning attacks, still maintaining 87.4% accuracy with label flipping and 40% independent malicious participants (robustness degrades under coordinated collusion, as detailed in Table 3), which is far better than FedAvg, which dropped to 53.2%. Future research directions will build on this framework to deal with real-time streaming threat intelligence circumstances adaptive privacy budgets depending on the intensity of the threats, multi-modal information rates, maliciously devolving methods in place of the best analytical processes, execution of cross-domain senior settling states of affairs, and boosting interpretability efficiencies to persuade legal adherence in teamed-up cybersecurity situations.

These results are demonstrated under the experimental conditions described in Section 4.1 and the 60-day pilot deployment described in Section 4.15. Broader applicability to other threat domains, larger federation sizes, or resource-constrained client environments requires additional validation.

Funding: The author received funding for this study from the Cybersecurity Research and Innovation Pioneers Initiative, provided by the National Cybersecurity Authority (NCA) in the Kingdom of Saudi Arabia (Grant No. CRPG-25-3036).

Data Availability Statement: The datasets used in this study are publicly available at: <https://www.unb.ca/cic/datasets/nsl.html> (accessed on 13 May 2026); <https://www.unb.ca/cic/datasets/ids-2017.html> (accessed on 13 May 2026); <https://www.unb.ca/cic/datasets/ids-2018.html> (accessed on 13 May 2026); <https://research.unsw.edu.au/projects/unsw-nb15-dataset> (accessed on 13 May 2026); <https://huggingface.co/datasets/joyce8/EMBER2024> (accessed on 13 May 2026). The Synthetic-CTI-2024 dataset was generated using CTGAN (embedding dim 128, three 256-unit hidden layers, 500 epochs, batch 500, Adam $\text{lr} = 2 \times 10^{-4}$), preserving statistical fidelity (MAE = 0.0032, Pearson RMSE = 0.041, KL-divergence = 0.0021; see Table 6). The NeuralFortress-XFL implementation includes HAFT ($L = 8$, $H = 8$), DPEM ($\epsilon = 2.5$, $\delta = 10^{-5}$), SAEP, and TWCM ($\beta_1 = 0.4$, $\beta_2 = 0.2$, $\beta_3 = 0.3$, $\beta_4 = 0.1$). Data partitioning employs Dirichlet ($\alpha = 0.5$) across 15 organizations with preprocessing (median imputation, winsorization, z-score normalization, k-anonymization $k = 5$). Random seeds are fixed at {42, 123, 456, 789, 1024}. The original CTI-Sharing-2024 dataset is restricted under FS-ISAC confidentiality agreements and cannot be released; however, all primary claims are reproducible using Synthetic-CTI-2024, which achieves 94.1% accuracy (vs. 95.6% for authentic data), confirming statistical fidelity (Table 6).

Conflicts of Interest: The author declares no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

Symbol/Acronym	Description
N	Number of participating organizations
D_i	Local dataset for organization i
n_i	Number of samples in organization i 's dataset
d	Feature dimensionality
θ	Global model parameters
$\theta_i(t)$	Local model parameters for organization i at round t
$\Delta\theta_i(t)$	Model update from organization i at round t
T	Total number of federated rounds
E	Number of local training epochs
ε	Privacy budget parameter
δ	Privacy slack parameter
$f\theta$	Threat detection model parameterized by θ
$\phi(x)$	Explanation for input x
$e_i(t)$	Explanation embedding for organization i at round t
$w_i(t)$	Trust score for organization i at round t
d_{model}	Transformer model dimensionality
L	Number of Transformer layers
H	Number of attention heads
d_{ff}	Feed-forward network dimensionality
d_e	Explanation embedding dimensionality
APT	Advanced persistent threat
CTI	Cyber threat intelligence
DDoS	Distributed denial of service
DPEM	Differential privacy explainability module
FL	Federated learning
HAFT	Hierarchical Attention-based Federated Transformer
LIME	Local Interpretable Model-agnostic Explanations
SAEP	Secure Aggregation with Explanation Preservation
TWCM	Trust-Weighted Contribution Mechanism
XAI	Explainable artificial intelligence

References

1. Prajwalasimha, S.N.; Shelke, N.; Saini, D.K.J.B.; Pimpalkar, A.; Kumar, G.H.; Shivamma, D. AI-powered defense against advanced persistent threats (APTs): Techniques, case studies, and future research directions. In *2025 3rd International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*; IEEE: Piscataway, NJ, USA, 2025; pp. 826–831. [\[CrossRef\]](#)
2. Pandey, S.; Azath, H.; Rahman, R.U.; Lamkuche, H. 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)*; IEEE: Piscataway, NJ, USA, 2025; pp. 437–442. [\[CrossRef\]](#)
3. Rahman, S.; Pal, S.; Jadidi, Z.; Karmakar, C. Robust cyber threat intelligence sharing using federated learning for smart grids. *IEEE Trans. Comput. Soc. Syst.* **2025**, *12*, 654–667. [\[CrossRef\]](#)
4. Camalan, E.; Celiktas, B. Privacy-preserving cyber threat intelligence: A framework combining private information retrieval, federated learning, and differential privacy. In *2025 10th International Conference on Computer Science and Engineering (UBMK)*; IEEE: Piscataway, NJ, USA, 2025; pp. 1525–1530. [\[CrossRef\]](#)
5. Alketbi, K.S.; Mehmood, A. A comprehensive survey of explainable artificial intelligence techniques for malicious insider threat detection. *IEEE Access* **2025**, *13*, 121772–121798. [\[CrossRef\]](#)
6. Jain, P.; Rathour, A.; Sharma, A.; Chhabra, G.S. Bridging explainability and security: An XAI-enhanced hybrid deep learning framework for IoT device identification and attack detection. *IEEE Access* **2025**, *13*, 127368–127390. [\[CrossRef\]](#)
7. Diop, M.M.; Ba, M.; Sylla, K.; Ouya, S. Artificial intelligence in cybersecurity: Applications, challenges, and future developments. In *2025 5th International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*; IEEE: Piscataway, NJ, USA, 2025; pp. 1–6. [\[CrossRef\]](#)

8. Durgaraju, S.; Vel, D.V.T.; Madathala, H. The evolution of cyber threats and defenses: A review of innovations and challenges. In *2025 6th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI)*; IEEE: Piscataway, NJ, USA, 2025; pp. 117–123. [\[CrossRef\]](#)
9. Mahir, S.H.; Tashrif, T.A.; Karim, A.; Kundu, D.; Rahman, A.; Hamza, A.; Al Farid, F.; Miah, A.S.M.; Mansor, S. Advanced hydro-informatic modeling through feedforward neural network, federated learning, and explainable AI for enhancing flood prediction. *IEEE Open J. Comput. Soc.* **2025**, *6*, 726–738. [\[CrossRef\]](#)
10. Bilal, M.A.; Islam, I.U.; Iltaf, N.; Khan, M.J.; Khan, M.J. Federated learning with explainable AI for malicious traffic detection in IoT networks. *IEEE Access* **2025**, *13*, 173368–173383. [\[CrossRef\]](#)
11. Alzamil, I. Federated deep learning for scalable and explainable load forecasting in privacy-conscious smart cities. *IEEE Access* **2025**, *13*, 3587058. [\[CrossRef\]](#)
12. Almadhor, E.A. 6G-enabled federated intelligence and transparent framework for aerial scene classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2025**, *18*, 25630–25639. [\[CrossRef\]](#)
13. Buyuktanir, B.; Altinkaya, Ş.; Baydogmus, G.K.; Yildiz, K. Federated learning in intrusion detection: Advancements, applications, and future directions. *Clust. Comput.* **2025**, *28*, 473. [\[CrossRef\]](#)
14. Chen, S.; Zheng, Q.; Long, Q.; Su, W.J. Minimax estimation for personalized federated learning: An alternative between FedAvg and local training? *J. Mach. Learn. Res.* **2023**, *24*, 1–59.
15. Cenitta, D.; Shwetha, G.K.; Rao, K.P.V.; Ramisetty, S.; Arul, N.; Arjunan, R.V. Explainable AI with homomorphic encryption for secure cloud-based ECG analysis in heart disease diagnosis. *IEEE Access* **2025**, *13*, 171831–171849. [\[CrossRef\]](#)
16. Asif, M.; Rehman, F.U.; Rashid, Z.; Hussain, A.; Mirza, A.; Qureshi, W.S. An insight on the timely diagnosis of diabetic retinopathy using traditional and AI-driven approaches. *IEEE Access* **2025**, *13*, 3583647. [\[CrossRef\]](#)
17. Bozorgpanah, A.; Torra, V. Explainable machine learning models with privacy. *Prog. Artif. Intell.* **2024**, *13*, 31–50. [\[CrossRef\]](#)
18. Fatema, K.; Anannya, M.; Dey, S.K.; Su, C.; Mazumder, R. Federated XAI IDS: An explainable and safeguarding framework with federated learning and SHAP in intrusion detection. *Future Internet* **2025**, *17*, 234. [\[CrossRef\]](#)
19. Alam, T.; Krishnaveni, B.V.; Babu, M.A.; Sallaah, M.H.; Pavithra, A.; K, A.S. Smart cities threat intelligence driven by artificial intelligence for enhanced cyber resilience using federated learning. In *2025 International Conference on Computational Innovations and Engineering Sustainability (ICCIES)*; IEEE: Piscataway, NJ, USA, 2025; pp. 1–6. [\[CrossRef\]](#)
20. Yan, N.; Wang, K.; Pan, C.; Chai, K.K.; Shu, F.; Wang, J. Over-the-air federated averaging with limited power and privacy budgets. *IEEE Trans. Commun.* **2024**, *72*, 1547–1561. [\[CrossRef\]](#)
21. Moshawrab, M.; Adda, M.; Bouzouane, A.; Ibrahim, H.; Raad, A. Reviewing federated learning aggregation algorithms: Strategies, contributions, limitations and future perspectives. *Electronics* **2023**, *12*, 2287. [\[CrossRef\]](#)
22. Jahan, S.; Adib, R.S.; Huda, S.M.; Rahman, S.; Kaiser, M.S.; Hosen, A.S.M.S.; Ghimire, D.; Park, M.J. Federated explainable AI-based Alzheimer’s disease prediction with multimodal data. *IEEE Access* **2025**, *13*, 3547343. [\[CrossRef\]](#)
23. Al Reshan, M.S.; Raza, M.O.; Mahoto, N.A.; Rajab, A.; Shaikh, A.; Elmagzoub, M.A.; Rajab, K.D. Toward drought modeling in South Asia: Machine learning approaches, challenges, and opportunities. *IEEE Access* **2025**, *13*, 3567257. [\[CrossRef\]](#)
24. Bhardwaj, T.; Sumangali, K. An explainable federated blockchain framework with privacy-preserving AI optimization for securing healthcare data. *Sci. Rep.* **2025**, *15*, 21799. [\[CrossRef\]](#) [\[PubMed\]](#)
25. Ribeiro, J.; Santos, R.; Analide, C.; Silva, F. Implementing federated learning and explainability techniques in regression models to increase transparency and reliability. *Stud. Inform. Control* **2024**, *33*, 15–24. [\[CrossRef\]](#)
26. Kasaraneni, B.P.; Putha, S.; Dunka, V.K.; Gudekota, S.; Pattyam, S.P.; Gayam, S.R.; Bhatt, M.W. Cyber-physical collaborative defense framework using artificial intelligence for cross-domain attacks in power systems. *IEEE Commun. Stand. Mag.* **2025**, early access. [\[CrossRef\]](#)
27. Cui, J.; Zong, L.; Xie, J.; Tang, M. A novel multi-module integrated intrusion detection system for high-dimensional imbalanced data. *Appl. Intell.* **2023**, *53*, 1027–1040.
28. Semenov, S.; Krupska-Klimczak, M.; Czapla, R.; Krzaczek, B.; Gavrylenko, S.; Poltorazkiy, V.; Vladislav, Z. Intrusion Detection Method Based on Preprocessing of Highly Correlated and Imbalanced Data. *Appl. Sci.* **2025**, *15*, 4243. [\[CrossRef\]](#)
29. Agrawal, S.; Sarkar, S.; Aouedi, O.; Yenduri, G.; Piamrat, K.; Alazab, M.; Bhattacharya, S.; Maddikunta, P.K.R.; Gadekallu, T.R. Federated Learning for Intrusion Detection System: Concepts, Challenges, and Future Directions. *Comput. Commun.* **2022**, *195*, 346–361. [\[CrossRef\]](#)
30. Mothukuri, V.; Parizi, R.M.; Pouriyeh, S.; Huang, Y.; Dehghantanha, A.; Srivastava, G. A Survey on Security and Privacy of Federated Learning. *Future Gener. Comput. Syst.* **2021**, *115*, 619–640. [\[CrossRef\]](#)
31. Friha, O.; Ferrag, M.A.; Shu, L.; Maglaras, L.; Choo, K.R.; Nafaa, M. FELIDS: Federated Learning-Based Intrusion Detection System for Agricultural Internet of Things. *J. Parallel Distrib. Comput.* **2022**, *165*, 17–31. [\[CrossRef\]](#)
32. Li, Y.; Chen, C.; Liu, N.; Huang, H.; Zheng, Z.; Yan, Q. A Blockchain-Based Decentralized Federated Learning Framework with Committee Consensus. *IEEE Netw.* **2021**, *35*, 234–241. [\[CrossRef\]](#)

33. Zhang, W.; Li, X.; Ma, H.; Luo, Z.; Li, X. Federated Learning for Machinery Fault Diagnosis with Dynamic Validation and Self-Supervision. *Knowl.-Based Syst.* **2021**, *213*, 106679. [[CrossRef](#)]
34. Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H.B.; Mironov, I.; Talwar, K.; Zhang, L. Deep Learning with Differential Privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16), New York, NY, USA, 24 October 2016; pp. 308–318. [[CrossRef](#)]
35. Liu, J.; Liu, X.; Wang, S.; Wan, X.; Li, D.; Lu, K.; He, K. Communication-efficient federated multi-view clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **2025**, *48*, 17–32. [[CrossRef](#)] [[PubMed](#)]
36. Dwork, C.; Roth, A. The Algorithmic Foundations of Differential Privacy. *Found. Trends Theor. Comput. Sci.* **2014**, *9*, 211–407. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.