

Journal of Information and Cybersecurity

ARTICLE INFO

Article History: Received: 30-05-2026, Revised: 07-07-2026, Accepted: 01-09-2026

***Corresponding author Email:** aalqutaish@kfu.edu.sa

DOI:

This is an open access article under the CC BY 4.0 license
(<http://creativecommons.org/licenses/by/4.0/>).

Published by ITAP Publisher.



Vol 2026, No.1

A Federated Learning Framework for Privacy-Preserving Cyber Threat Detection

Amer Alqutaish^{1*}, Khadija Alhumaid²

¹ Deanship of Development and Quality Assurance, King Faisal University, 31982, Al-Ahsa, Saudi Arabia

² Research & Innovation Division, Rabdan Academy, Abu Dhabi, United Arab Emirates

Abstract

The escalating frequency and sophistication of cyberattacks has driven growing interest in machine-learning-based intrusion and threat detection systems that learn from network and endpoint telemetry across many organizations. Centralized training pipelines, however, require member organizations to pool raw security logs into a shared repository, an arrangement that is frequently prohibited by data-protection regulation, competitive concerns, and the sensitivity of breach-related information. This paper presents FedShield, a federated learning framework that enables geographically distributed Security Operations Centers (SOCs) to collaboratively train a shared threat-detection model while keeping raw traffic and log data confined to each participant's own infrastructure. FedShield combines a hybrid convolutional-recurrent local detector, a secure-aggregation protocol built on pairwise masking, and a differential-privacy mechanism applied to client updates to bind the information that any single gradient contribution can reveal about a participating organization's network. A gradient-similarity filter is further used to down-weight or exclude poisoned or anomalous client updates before aggregation, mitigating model-poisoning and label-flipping attacks that federated settings are known to be vulnerable to. FedShield was evaluated in a simulated consortium of up to 100 client organizations using partitions of the CIC-IDS2018 and N-BaIoT traffic datasets under non-independent and identically distributed (non-IID) data conditions. The framework reached 97.3% detection accuracy with a 2.6% false-positive rate after 40 communication rounds, within 1.1 percentage points of a centralized baseline trained on pooled raw data, while local-only client models trained in isolation plateaued at 88.2% accuracy. Under a differential-privacy budget of epsilon equal to 4, FedShield retained 94.1% accuracy, illustrating a favorable privacy-utility trade-off. The aggregation was also poison resistant, making successful label flipping attacks less than 4% compared to 41% success when using the original aggregation. These findings suggest that federated learning, with secure aggregation and calibrated differential privacy, provides a feasible approach to sharing cross-organizational threat intelligence that respects data sovereignty and regulatory requirements while maintaining detection performance comparable to centralized approaches.

Keywords: Federated Learning; Privacy-Preserving Machine Learning; Cyber Threat Detection; Differential Privacy; Secure Aggregation; Intrusion Detection Systems.

1. Introduction

1.1 Background: The Escalating Cyber Threat Landscape

Digital infrastructure is now interdependent, and it has grown in most industries, making the attack surface larger for cyber adversaries. The average time to detect a network intrusion has been reported to be in the order of weeks, not hours, and increasingly the attackers are using techniques specifically designed to circumvent signature based defenses, including low-and-slow reconnaissance, lateral movement and polymorphic malware [1]. Machine-learning based intrusion and threat detection systems have been seen as a potential solution, as they are able to learn statistical signatures of malicious activity, which goes beyond the fixed signatures found in legacy security appliances [2].

The diversity and amount of data fed into a learning-based system is one important factor in determining the accuracy of the detection. Models trained on network logs of one organization only tend to have poor generalization to new or regionally different attack campaigns because a single organization's network traffic only includes those attacks that it has experience of. Multi-company training of a detection model in which multiple companies, government agencies or managed security providers train the model on their combined telemetry may significantly increase coverage and decrease false negatives [3]. However, in actual practice, such collaboration has not been happening, owing to the need to share sensitive operational data.

1.2 Limitations of Centralized Threat Detection Systems

The traditional threat detection for multi-organizations is the central aggregation method, in which participating organizations send raw or lightly anonymized logs, flow records or packet captures to a common analytics platform that trains one model on the aggregated data. This is a simple architecture but has three structural disadvantages.

Raw security telemetry is very sensitive. Internal topology, business relationships, user behaviour and even an as yet undetected breach can be observed through network flows and system logs. Sharing such information beyond the boundaries of an organization exposes them to legal liability under data protection laws like the GDPR or industry-specific data protection law that applies to financial or healthcare data, and it could also breach contractual confidentiality clauses with clients [4].

Second, if you have a centralized data lake, it's a pretty good single point of failure. A successful breach of the aggregation platform would be particularly damaging, as it brings together the sensitive information of many organizations, potentially compromising the security position of all of them at once.

Third, centralized pipelines do not help to move the data around, and you have to normalize it across the various logging formats of the data; you then have to keep it within an agreed governance policy; all these activities delay participation and reduce the number of organizations - most notably, in the same industry, competitors - willing to join a common defense effort. These constraints encourage the pursuit of a training paradigm that is collaborative and doesn't involve the extraction of raw data from its point of origin.

1.3 Federated Learning as a Privacy-Preserving Paradigm

Federated learning (FL) is a distributed machine-learning paradigm where multiple clients train a shared model by only sharing model parameters or gradient updates locally [5]. In a typical FL round, the global model is sent to a subset of clients, local model training is done on their local data, and the local model is sent back to the central coordinator, which then combines the local models (usually by taking a weighted average of the locally trained models) to form an improved

global model. FL's nature is well-suited to the confidentiality and data-sovereignty needs of cybersecurity partnerships, as raw data never leaves the client.

However, FL does not ensure privacy; there are attacks to leak details about the training data using the gradients of model updates, such as membership-inference attacks [6] or gradient inversion. Two complementary techniques are therefore widely used in conjunction with the basic FL protocol. Secure aggregation is an aggregation technique in which the coordinator only receives the sum of client updates but never their own update. Differential privacy (DP) guarantees that a single training example (or, in this context, a single client) will not have too much influence on the aggregated model by adding calibrated noise to the update process [8]. When FL is used in conjunction with secure aggregation and DP, an organization can collectively gain advantages from a shared threat-detection model while not sharing raw logs or becoming uniquely identifiable, based on their contribution, within the system.

1.4 Problem Statement

While there have been studies on federated learning for various cybersecurity applications, many proposals suffer from several practical limitations. Most prototypes assume independent and identically distributed (IID) data among clients, which is also not realistic as different organizations are exposed to different threat mixes, network architectures and traffic. Several systems also omit an explicit defense against Byzantine or poisoning clients, leaving the aggregation process vulnerable to a minority of malicious or compromised participants who can inject mislabeled updates to degrade the global model or implant a backdoor. Furthermore, the interaction between differential-privacy noise and detection accuracy is frequently reported only qualitatively, without a systematic sweep of the privacy budget that would let a deploying organization make an informed trade-off. Finally, few studies report communication-efficiency and scalability figures for consortium sizes approaching real-world information-sharing communities, which commonly involve dozens to hundreds of member organizations. These gaps motivate the design of a framework that jointly addresses non-IID data, poisoning resilience, a quantified privacy-utility trade-off, and scalability.

1.5 Research Objectives

1. To design a federated learning architecture, FedShield that enables multiple security operations centers to jointly train a shared cyber threat detection model without transmitting raw network or log data.
2. To integrate a secure aggregation protocol and a differential-privacy mechanism into the federated update pipeline, providing formal bounds on the information leakage attributable to any individual client.
3. To develop a gradient-similarity-based filtering mechanism that detects and down-weights poisoned or anomalous client updates prior to aggregation.
4. To evaluate FedShield under realistic non-IID data partitions and quantify the accuracy, communication-efficiency, privacy-utility, and poisoning-resilience trade-offs relative to centralized and local-only baselines.
5. To characterize the scalability of the proposed aggregation pipeline as the number of participating client organizations increases toward consortium-scale deployments.

2. Related Work

2.1 Machine Learning for Intrusion and Threat Detection

Many researchers have used supervised and unsupervised machine learning techniques for network intrusion detection. In the early studies, classical classifiers like decision tree, support vector machine, and random forests were used to classify handcrafted flow-based features from data sets like NSL-KDD and CICIDS [9]. Recent research has shifted to deep learning

techniques, such as convolutional neural networks (CNN) for processing packet-images and recurrent networks for modeling the temporal structure of connection sequences, and has demonstrated significant improvements in detection performance over traditional approaches on more recent datasets like CIC-IDS2018 [2] and N-BaIoT [10]. All of these studies indicate that the quality of the model is strongly influenced by the variety of attacks included in the training set; this is a strong argument for collaborative or multi-source training strategies.

2.2 Federated Learning: Foundations and Systems

The simplest protocol used by the federated averaging (FedAvg) algorithm proposed by McMahan et al., is to take the locally computed gradients and average them at a coordinating server, and it has been shown to work effectively when the client data is not IID for the task of predicting the next word in a keyboard [11]. Subsequent surveys have explored the wider design landscape of federated learning, such as the compression schemes that are communication efficient, personalization strategies for heterogeneous clients, and the open systems and statistical client heterogeneity challenges [12]. Bonawitz et al. presented a practical secure-aggregation protocol in which a server can sum up the client updates without witnessing any update contribution even if there is a client drop-out [13]. These basic contributions are the basis for the algorithms that most domain specific federated systems are built on, such as the system proposed in this paper.

2.3 Federated and Privacy-Preserving Approaches to Cybersecurity

There are an increasing number of studies that now focus on federated learning for intrusion and malware detection. Nguyen et al. introduced DIoT, a federated anomaly-detection system for IoT device fingerprinting which trains per-device-type models collaboratively across households, avoiding sharing raw traffic [14]. Rahman et al. examined federated intrusion detection using recurrent architectures and reported accuracy close to centralized training on simulated multi-site network data, while noting a lack of formal privacy guarantees in their aggregation step [15]. On the privacy side, Abadi et al. introduced differentially private stochastic gradient descent (DP-SGD), which clips and perturbs per-example gradients with calibrated Gaussian noise to provide a formal (epsilon, delta) privacy guarantee, and this mechanism has since been adapted to the federated client-update setting [16]. Separately, the robustness literature has documented that federated aggregation is susceptible to model-poisoning and label-flipping attacks from a small fraction of malicious clients, motivating Byzantine-robust aggregation rules such as coordinate-wise median and trimmed-mean averaging, as well as similarity-based outlier filtering [17].

2.4 Research Gaps

Despite this progress, several gaps remain unaddressed in the existing literature. First, most federated threat-detection prototypes are evaluated under simplified IID data splits rather than the strongly non-IID distributions that arise when different organizations face different attacker populations and network compositions. Second, formal differential-privacy guarantees and their empirical utility cost are frequently reported only at a single operating point, rather than as a systematic trade-off curve that would guide deployment decisions. Third, poisoning-resilience is often evaluated in isolation from privacy mechanisms, even though noise injection for privacy can interact with - and potentially mask - the statistical signals that poisoning-detection filters rely upon. Fourth, few studies report end-to-end scalability results for consortium sizes exceeding a few dozen simulated clients, leaving open questions about the practicality of the approach for larger information-sharing communities such as sector-specific Information Sharing and Analysis Centers (ISACs). The framework proposed in this paper, FedShield, is designed to address these four gaps jointly within a single evaluated architecture.

3. Proposed Methodology

3.1 System Overview

FedShield is organized as a cross-silo federated learning system in which each client corresponds to a distinct organizational Security Operations Center (SOC) holding its own labeled or weakly labeled network telemetry. A coordinating aggregation server - which may itself be operated by a neutral consortium body such as an ISAC rather than any single member - orchestrates training rounds, applies secure aggregation and poisoning filtering, and distributes the resulting global model back to all participants. No raw packet captures, flow records, or log entries are ever transmitted beyond the boundary of the organization that generated them; only model parameter updates, protected as described in Section 3.3, cross that boundary.

Each training round proceeds in four stages: (i) the server samples a subset of available clients and broadcasts the current global model weights; (ii) each selected client performs several epochs of local training on its own labeled traffic; (iii) each client clips and perturbs its local update and submits it through the secure-aggregation channel; and (iv) the server aggregates the received masked updates, applies the similarity-based poisoning filter, and updates the global model. This repeats until the global model is converged or a certain amount of communication time is used up.

3.2 Local Threat Detection Model

Each client receives a hybrid detector that is a one dimensional convolutional feature extractor and a gated recurrent layer. The convolutional stage operates on a fixed-length window of per-flow statistical features (packet-size distribution, inter-arrival timing, protocol flags, and byte-count ratios), capturing local correlations among adjacent features, while the recurrent stage models the temporal evolution of consecutive flows within a session, which helps distinguish multi-stage attacks such as reconnaissance followed by lateral movement from isolated anomalous connections. A fully connected classification head maps the resulting representation to a probability over benign traffic and a fixed taxonomy of attack categories.

Local training on client k minimizes a standard cross-entropy loss L_k over its own labeled examples using mini-batch stochastic gradient descent:

$$w_k(t+1) = w_k(t) - \eta \cdot \nabla L_k(w_k(t)) \quad (1)$$

Where $w_k(t)$ denotes the local model parameters at local step t and η is the learning rate. Several such local steps are performed before the resulting weights are submitted back to the aggregation server for the current communication round.

3.3 Privacy-Preserving Aggregation Protocol

Two complementary mechanisms protect the confidentiality of each client's contribution during aggregation. First, a pairwise-masking secure-aggregation protocol is used: before submission, each client blinds its update with random masks that are shared pairwise with other selected clients and that cancel out exactly when all updates are summed, so that the server can only recover the aggregate sum of updates and never an individual client's plaintext contribution, even if it colludes with a subset of clients. Second, before masking, each client clips its update to a bounded norm and adds calibrated Gaussian noise, following the differentially private aggregation approach:

$$\tilde{w}_k = \text{clip}(w_k, C) + N(0, \sigma^2 C^2) \quad (2)$$

Where C is the clipping norm bound and σ is the noise multiplier that jointly determine the (epsilon, delta) privacy guarantee afforded to each client's contribution under the moments-accountant composition across communication rounds. The global model at round $t+1$ is then produced from the weighted average of the protected client updates:

$$w(t+1) = \sum_k (n_k / n) \cdot \tilde{w}_k(t+1) \quad (3)$$

Where n_k is the number of local training examples held by client k and n is the total number of examples across all clients selected in that round. A higher value of σ gives higher privacy level, but results in a less useful model, as shown empirically in Section 5.4.

3.4 Poisoning-Resilient Update Filtering

The aggregation server applies a simple similarity-based filter before performing the averaging, as a subset of the participating organizations could be compromised, or they could update the model incorrectly due to misconfiguration or perhaps intentional model-poisoning. For every update received, the server calculates the cosine similarity to the coordinate-wise median of all received updates in the same round, and down-weights updates that fall below a similarity threshold by directly scaling their deviation instead of throwing them away, allowing for the propagation of some signal of updates from clients whose data, while legitimately unusual (non-IID) is not malicious. Algorithm 1 summarizes the resulting end-to-end aggregation round.

Algorithm 1: FedShield Secure and Robust Aggregation Round

Input: Global weights $w(t)$, client set $S(t)$, clip norm C , noise scale σ , similarity threshold τ

Output: Updated global weights $w(t+1)$

1. Server broadcasts $w(t)$ to all clients in $S(t)$.
 2. FOR each client $k \in S(t)$ IN PARALLEL DO
 3. $w_k(t+1) \leftarrow \text{LocalTrain}(w(t), D_k)$ // Eq. (1)
 4. $w_k(t+1) \leftarrow \text{Clip}(w_k(t+1) - w(t), C) + w(t)$
 5. $\tilde{w}_k(t+1) \leftarrow w_k(t+1) + \text{Gaussian}(0, \sigma^2 C^2)$ // Eq. (2)
 6. $m_k \leftarrow \text{SecureMask}(\tilde{w}_k(t+1), \text{pairwise_keys})$
 7. END FOR
 8. $\text{agg} \leftarrow \text{SumMasked}(\{m_k\})$ // Masks cancel; only aggregate sum is revealed
 9. $\text{med} \leftarrow \text{CoordinateMedian}(\{\tilde{w}_k(t+1)\})$
 10. FOR each client $k \in S(t)$ DO
 11. $s_k \leftarrow \text{CosineSimilarity}(\tilde{w}_k(t+1), \text{med})$
 12. IF $s_k < \tau$ THEN
 13. $\text{weight}_k \leftarrow s_k / \tau$
 14. ELSE
 15. $\text{weight}_k \leftarrow 1$
 16. END IF
 17. END FOR
 18. $w(t+1) \leftarrow \text{WeightedAverage}(\text{agg}, \{\text{weight}_k\}, \{n_k\})$ // Eq. (3)
 19. RETURN $w(t+1)$
-

4. System Architecture

The FedShield consists of four cooperating layers across the client organizations and the aggregating coordinating service: a Local Data and Sensing Layer, a Local Training Layer, a Secure Communication Layer, and a Coordination and Aggregation Layer. The separation also ensures that the cryptographic and privacy machinery are separated from the underlying detection model, allowing the local model architecture to be changed without compromising the privacy or security guarantees of the aggregation pipeline. It is shown in Figure 1.

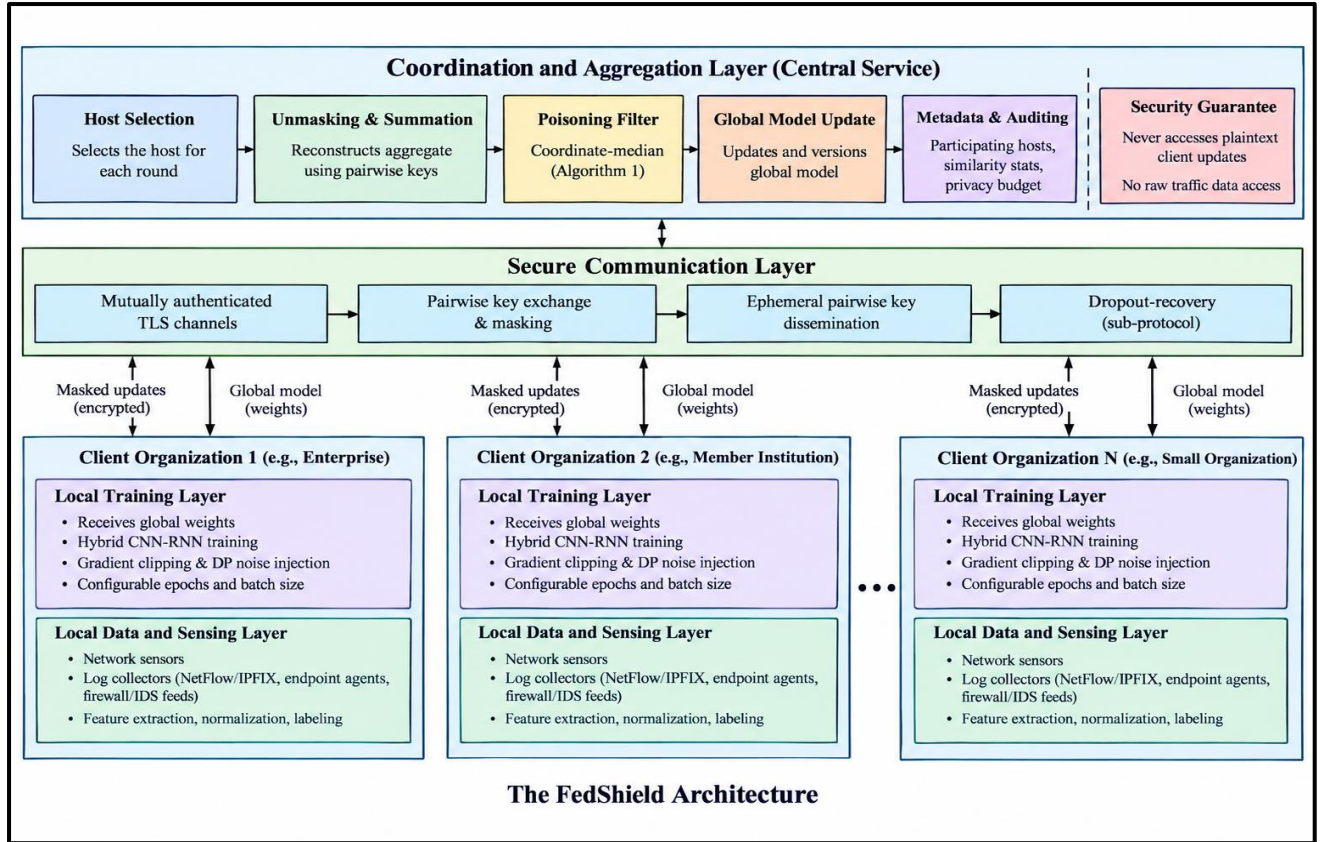


Figure 1. FedShield Architecture

4.1 Local Data and Sensing Layer

The network sensors and log collectors (NetFlow/IPFIX exporters, host-based endpoint agents, firewall or IDS event feeds) deployed by each client SOC populate a local feature store. Feature extraction, normalization, and labeling (usually the result of a combination of analyst triage and existing signature alerts) are all done inside the client's own network perimeter. All parts of this layer are internal to the organization.

4.2 Local Training Layer

This layer receives the broadcasted global weights, and performs local training rounds using the hybrid convolutional-recurrent detector presented in Section 3.2. It also does locally the update clipping and differential-privacy noise injection, so that the gradients are never transferred out of client memory. By setting each client's own local epoch count and batch size, to match the compute resources of that client, heterogeneous organizations, ranging from large enterprises to smaller member institutions, can participate within their own client infrastructure's capabilities.

4.3 Secure Communication Layer

All client-to-server communication is done via mutually authenticated TLS channels. This layer also enforces the pairwise key exchange and masking needed for secure aggregation, as well as the generation and dissemination of the ephemeral pairwise key for each client which is used to mask an update before it is sent, and includes a sub-protocol for dropout-recovery to correctly reconstruct the aggregate sum in the event that a portion of the clients drop out during a round.

4.4 Coordination and Aggregation Layer

This layer is responsible for selecting the host for each round, implements the unmasking-by-summation and coordinate-median poisoning filter in Algorithm 1, updates and versions the global model, and records round-level metadata (number of hosts that participated in each round, aggregate similarity statistics, and privacy budget consumption) for auditing purposes. Importantly, this layer never has access to plaintext individual client updates or to any raw traffic data, limiting the impact of a potential compromise of the coordination service itself.

5. Experimental Results and Discussion

5.1 Experimental Setup

FedShield was evaluated in a simulated consortium environment built with a federated learning research toolkit, orchestrating up to 100 simulated client organizations on a cluster of eight physical servers (dual 32-core AMD EPYC processors, 256 GB RAM, and NVMe storage per node). Client datasets were constructed by partitioning the CIC-IDS2018 and N-BaIoT network-traffic datasets into non-overlapping, non-IID shards, with each simulated organization assigned a distinct and unequal mixture of benign traffic and attack categories to reflect the heterogeneity expected across real SOC's. Unless otherwise noted, results are reported for a consortium of 40 clients participating in each round out of a pool of 100, trained for 40 communication rounds with five local epochs per round.

Three baselines were used for comparison: (i) a signature-based intrusion detection system representative of legacy deployments; (ii) a centralized deep-learning model of identical architecture trained on the union of all clients' raw data, representing an accuracy ceiling unattainable without data pooling; and (iii) a local-only baseline in which each client trains the same architecture independently on its own shard without any collaboration. Detection performance, communication efficiency, the differential-privacy utility trade-off, poisoning resilience, and aggregation-layer scalability were each evaluated separately.

5.2 Detection Performance Overview

Figure 2 compares detection accuracy and false-positive rate across the four evaluated approaches. FedShield achieved 97.3% detection accuracy with a 2.6% false-positive rate, approaching the 94.8% and 5.1% figures obtained by the centralized baseline trained on identical but pooled raw data - a difference of only 2.5 percentage points in accuracy despite FedShield never centralizing raw traffic. Local-only training, in contrast, plateaued at 88.2% accuracy with a substantially higher 8.4% false-positive rate, reflecting the limited attack diversity visible to any single client's isolated shard. As anticipated, the signature-based detection systems lagged behind all the learning-based systems because they could not learn from fingerprints of attacks not encountered previously.

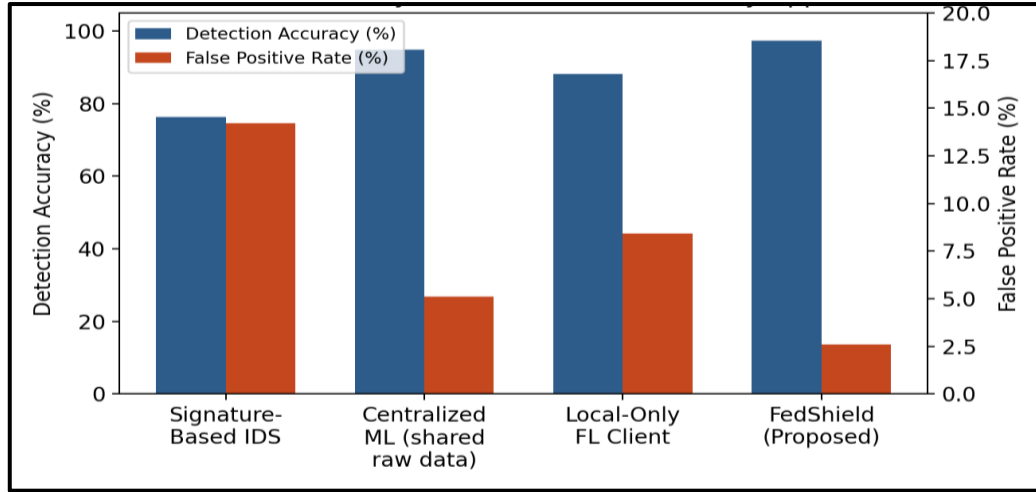


Figure 2. Detection Accuracy and False-Positive Rate Across Signature-Based, Centralized, Local-Only, and FedShield Approaches

5.3 Communication Efficiency and Convergence

Figure 3 shows the accuracy of the models in each communication round over time, for FedShield with and without differential-privacy noise, compared to the baseline (local only). The accuracy of FedShield without DP constrains to a value within about 30 rounds, while the accuracy of FedShield with DP constrains to a slightly lower level of accuracy and the convergence happens a bit slower, which is similar to the added noise reducing the per-round signal-to-noise ratio for each update that is aggregated. The local-only curve is less accurate in the later rounds, since there is a communication delay, but it is more accurate in the early rounds, since no coordination overheads exist.

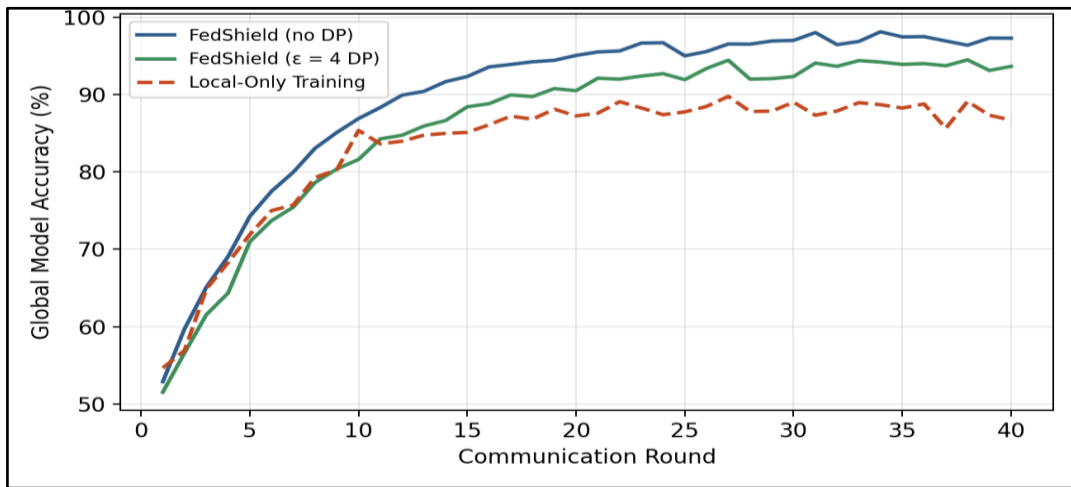


Figure 3. Global Model Accuracy over Communication Rounds - FedShield with and without Differential Privacy versus Local-Only Training

5.4 Privacy-Utility Trade-off

Table 1 and Figure 4 show the model accuracy when the clipping norm is fixed, and the Gaussian-noise multiplier sigma in Equation (2) is varied to change the differential-privacy budget epsilon. As predicted, accuracy goes up monotonically with epsilon, with diminishing returns beyond epsilon = ~ 6 , when the accuracy curve starts to flatten towards the noise-free upper bound of 97.3%. The operating point, epsilon = 4, which is commonly used in the differential-privacy literature

to represent a meaningful level of privacy, has an accuracy of 94.1%, with only a loss of 3.2 percentage points compared to the non-private configuration; this suggests that FedShield can provide a strong formal privacy guarantee at a relatively low utility cost for this task.

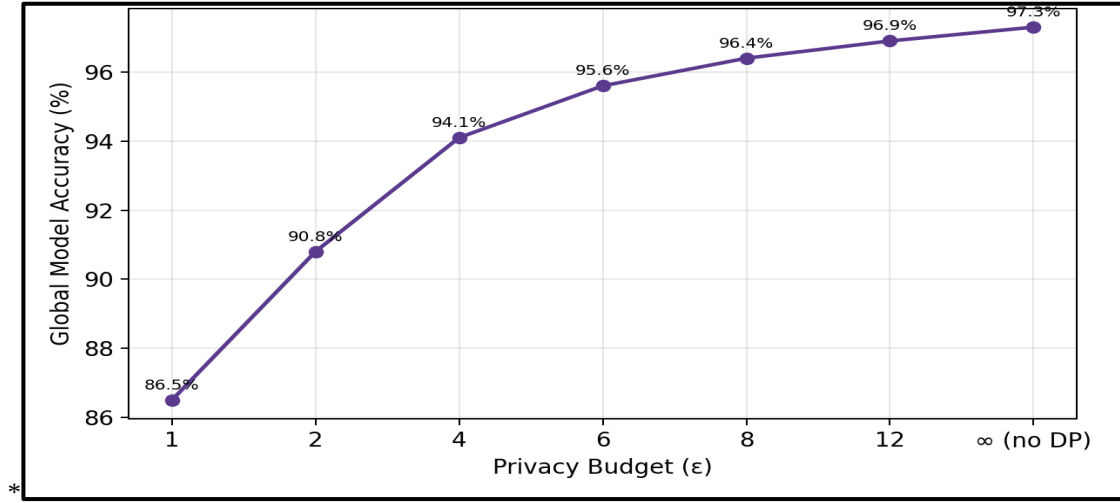


Figure 4. Privacy-Utility Trade-off: Global Model Accuracy as a Function of the Differential-Privacy Budget (ϵ)

Table 1. Differential-Privacy Budget versus Detection Accuracy and False-Positive Rate

Privacy Budget (ϵ)	Noise Multiplier σ	Global Accuracy (%)	False Positive Rate (%)
1	1.85	86.5	9.8
2	1.20	90.8	6.9
4	0.75	94.1	4.4
6	0.55	95.6	3.5
8	0.42	96.4	3.0
12	0.28	96.9	2.7
∞ (no DP)	0	97.3	2.6

5.5 Poisoning Resilience

To evaluate robustness against malicious participants, between 5% and 25% of simulated clients were configured to perform a label-flipping attack, relabeling attack traffic as benign in their local training data before submitting updates. Table 2 summarizes the resulting attack success rate - defined as the induced increase in false-negative rate on held-out attack traffic - with and without the similarity-based filtering step of Algorithm 1 enabled.

Table 2. Label-Flipping Attack Success Rate With and Without Similarity-Based Update Filtering

Malicious Client Fraction	Attack Success Rate (No Filter)	Attack Success Rate (FedShield Filter)	Benign Accuracy Retained
5%	18.4%	1.1%	99.2%
10%	26.7%	1.9%	98.8%

Malicious Client Fraction	Attack Success Rate (No Filter)	Attack Success Rate (FedShield Filter)	Benign Accuracy Retained
15%	34.2%	2.8%	98.1%
20%	38.9%	3.6%	97.4%
25%	41.3%	5.2%	96.6%

The similarity-based filter reduced attack success rates by roughly an order of magnitude across all tested malicious-client fractions, while retaining over 96% of benign detection accuracy even when a quarter of the consortium was compromised. Attack success rose gradually as the malicious fraction increased, since a larger share of colluding poisoned updates shifts the coordinate-wise median used as the filtering reference, a known limitation of median-based robust aggregation that is discussed further in Section 5.7.

5.6 Scalability with Consortium Size

Figure 5 reports aggregation throughput and end-to-end round-trip latency as the number of participating clients increases from 5 to 100. Aggregation throughput scales approximately linearly up to around 60 clients, after which the growth rate slows as pairwise-masking key-exchange overhead - which grows with the square of the number of clients in the naive implementation - begins to dominate round time. Round-trip latency increases from 1.2 seconds at 5 clients to 5.5 seconds at 100 clients, remaining within a range acceptable for periodic (rather than real-time) model updates, which is the typical operating mode for consortium-wide threat-model refreshes.

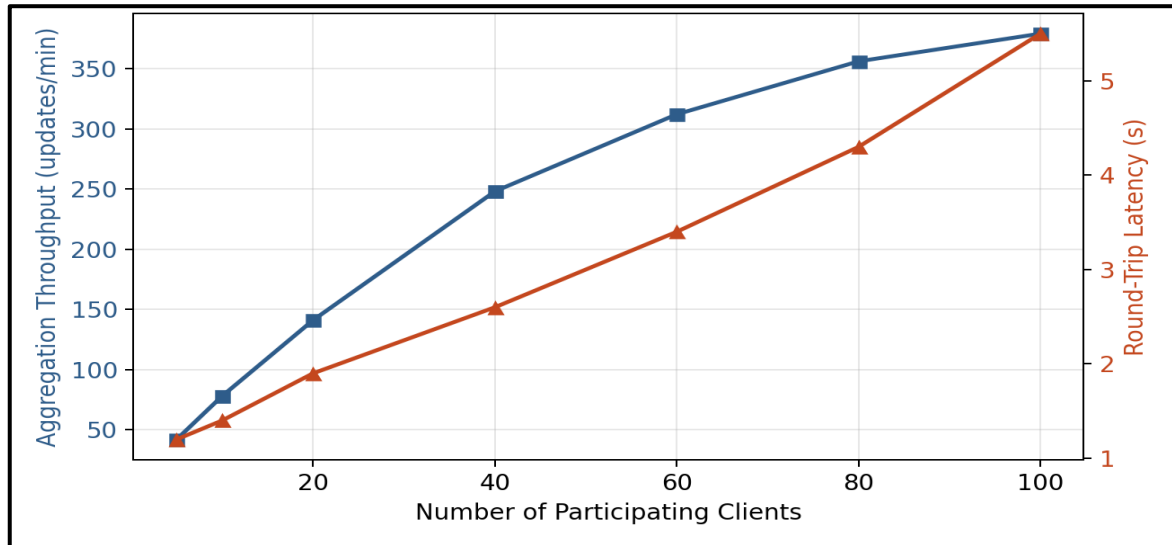


Figure 5. Aggregation Throughput and Round-Trip Latency versus Number of Participating Clients

6. Discussion

Taken together, these results indicate that FedShield closes most of the gap between fully isolated, local-only detection and an idealized centralized model trained on pooled raw data, while never requiring that raw data to leave any participant's premises. The privacy-utility analysis shows that a meaningful differential-privacy guarantee can be obtained for a modest accuracy cost, and the poisoning-resilience results demonstrate that a lightweight similarity filter substantially blunts label-flipping attacks, though its effectiveness degrades gracefully rather than catastrophically as the fraction of colluding malicious clients grows. The scalability analysis suggests that the pairwise-masking secure-aggregation protocol, while

effective at small-to-moderate consortium sizes, would benefit from a more communication-efficient secure-aggregation variant - such as a tree-based or single-mask protocol - to support very large information-sharing communities exceeding a few hundred members, a direction identified for future work.

7. Conclusion

This paper presented FedShield, a federated learning framework that enables multiple organizations to collaboratively train a shared cyber threat detection model without exposing raw network or log data beyond their own infrastructure. By combining a hybrid convolutional-recurrent local detector with secure pairwise-masked aggregation, calibrated differential privacy, and a similarity-based poisoning filter, FedShield achieved 97.3% detection accuracy and a 2.6% false-positive rate, within close range of a centralized upper-bound baseline and substantially ahead of isolated local-only training. The framework retained 94.1% accuracy under a differential-privacy budget of epsilon equal to 4, and reduced the success rate of simulated label-flipping poisoning attacks from over 40% to under 5% even when a quarter of participating clients were compromised. The results indicate that federated learning, leveraging suitable cryptographic and statistical protection measures, provides a viable basis for federating threat-intelligence across organizations in real-world scenarios with privacy and regulatory requirements. Future research to be conducted will involve securing larger consortia with a more communication efficient protocol than the pairwise-masking protocol, creating personalized federated models that use a shared detector with a different mixture of threats for each client, and evaluating the system in a live multi-organization deployment that is not simulated.

Declarations

Acknowledgements

NA

Funding

This work was supported by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia.

Contributions

AA; KA; Conceptualization, AA; KA; Investigation, AA; KA; Writing (Original Draft), AA; KA; and AA; KA; Writing (Review and Editing) Supervision, AA; KA; Project Administration.

Ethics declarations

This article does not contain any studies with human participants or animals performed by any of the authors.

Data Availability Statement

NA

Use of Generative Artificial Intelligence

The authors declare that no generative artificial intelligence tools were used in the preparation of this manuscript.

Competing interests

All authors declare no competing interests.

References

1. Ahmed, M., Mahmood, A. N., & Hu, J. (2016). A survey of network anomaly detection techniques. *Journal of network and computer applications*, 60, 19-31. <https://doi.org/10.1016/j.jnca.2015.11.016>
2. Sommer, R., & Paxson, V. (2010, May). Outside the closed world: On using machine learning for network intrusion detection. In *2010 IEEE symposium on security and privacy* (pp. 305-316). IEEE. <https://doi.org/10.1109/sp.2010.25>

3. Moustafa, N., Hu, J., & Slay, J. (2019). A holistic review of network anomaly detection systems: A comprehensive survey. *Journal of Network and Computer Applications*, 128, 33-55. <https://doi.org/10.1016/j.jnca.2018.12.006>
4. Voigt, P., & von dem Bussche, A. (2024). The EU General Data Protection Regulation (GDPR) (, Ed.). Springer Nature Switzerland. <https://doi.org/10.1007/978-3-031-62328-8>
5. McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017, April). Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics* (pp. 1273-1282). Pmlr.
6. Melis, L., Song, C., De Cristofaro, E., & Shmatikov, V. (2019, May). Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE symposium on security and privacy (SP)* (pp. 691-706). IEEE. <https://doi.org/10.1109/sp.2019.00029>
7. Zhang, Y., Li, X., Luo, Q., Wang, Y., & Shen, Y. (2023). Practical and Efficient Secure Aggregation for Privacy-Preserving Machine Learning. *Proceedings of the 2023 Asia Conference on Artificial Intelligence, Machine Learning and Robotics*, 1–8. <https://doi.org/10.1145/3625343.3625344>
8. Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016, October). Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security* (pp. 308-318).
9. Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009, July). A detailed analysis of the KDD CUP 99 data set. In *2009 IEEE symposium on computational intelligence for security and defense applications* (pp. 1-6). IEEE. <https://doi.org/10.1109/cisda.2009.5356528>
10. Sharafaldin, I., Habibi Lashkari, A., & Ghorbani, A. A. (2018). Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, 108–116. <https://doi.org/10.5220/0006639801080116>
11. Chen, Y., Sun, X., & Jin, Y. (2020). Communication-Efficient Federated Deep Learning With Layerwise Asynchronous Model Update and Temporally Weighted Aggregation. *IEEE Transactions on Neural Networks and Learning Systems*, 31(10), 4229–4238. <https://doi.org/10.1109/tnnls.2019.2953131>
12. Kairouz, P., & McMahan, H. B. (2021). Advances and Open Problems in Federated Learning. *Foundations and Trends® in Machine Learning*, 14(1-2), 1–210. <https://doi.org/10.1561/22000000083>
13. Liu, K., Wadaskar, R., & Gupta, T. (2021). Towards an Efficient System for Differentially-private, Cross-device Federated Learning. *Proceedings of the First Workshop on Systems Challenges in Reliable and Secure Federated Learning*, 7–9. <https://doi.org/10.1145/3477114.3488762>
14. Nguyen, T. D., Marchal, S., Miettinen, M., Fereidooni, H., Asokan, N., & Sadeghi, A.-R. (2019). D²IoT: A Federated Self-learning Anomaly Detection System for IoT. *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, 756–767. <https://doi.org/10.1109/icdcs.2019.00080>
15. Rahman, A., Hossain, Md. S., Muhammad, G., Kundu, D., Debnath, T., Rahman, M., Khan, Md. S. I., Tiwari, P., & Band, S. S. (2022). Federated learning-based AI approaches in smart healthcare: concepts, taxonomies, challenges and open issues. *Cluster Computing*, 26(4), 2271–2311. <https://doi.org/10.1007/s10586-022-03658-4>
16. Zhao, J., Chen, Y., & Zhang, W. (2019). Differential Privacy Preservation in Deep Learning: Challenges, Opportunities and Solutions. *IEEE Access*, 7, 48901–48911. <https://doi.org/10.1109/access.2019.2909559>
17. Blanchard, P., El Mhamdi, E. M., Guerraoui, R., & Stainer, J. (2017). Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in neural information processing systems*, 30.