Innovative Journal of Computing, Security and Digital Technologies
Volume 1, Issue 1, Jan – Apr 2026
Page no : 15-22

Privacy-Preserving Federated Learning for Intrusion Detection in Distributed Networks

Shahbad Daulatpur

Department of Computer Science and Engineering, Delhi Technological University, Delhi–110042, India.

Abstract

Distributed networks generate security data across organizations, cloud platforms, edge devices, and Internet of Things (IoT) environments. Centralizing this data for intrusion detection can create privacy, governance, communication, and operational challenges. Federated learning (FL) enables multiple participants to collaboratively train a detection model while retaining raw traffic records locally. However, keeping data local does not by itself guarantee privacy: exchanged model updates may reveal information, and malicious participants may poison the training process. This paper presents a conceptual privacy-preserving federated learning framework for distributed intrusion detection. The proposed workflow combines local data preparation, collaborative model training, secure aggregation, differential privacy, update validation, and continuous model evaluation. It also considers non-identically distributed data, communication overhead, client availability, and the trade-off between privacy protection and detection utility. The paper is design-oriented and does not report experimental results. It outlines an evaluation plan for future validation using distributed network datasets and realistic adversarial scenarios.

Keywords— Federated learning, intrusion detection, distributed networks, differential privacy, secure aggregation, network security, privacy preservation.

Publication Date : 20.01.2026

I. INTRODUCTION

Distributed networks connect users, devices, applications, and services across multiple administrative and technical environments. Their scale and diversity make intrusion detection challenging: malicious activity may appear differently across network segments, and each participant may hold only a partial view of the overall threat landscape.

Conventional centralized intrusion detection systems (IDSs) collect network records at a central location for analysis. This can simplify model training but may require sensitive traffic information to leave the network where it was generated. Such transfer can raise privacy, compliance, bandwidth, and data-governance concerns. Recent reviews identify federated learning as a promising approach for collaborative intrusion detection while also highlighting unresolved challenges involving privacy, data heterogeneity, and security of model updates [1], [2].

How to Cite
Privacy-Preserving Federated Learning for Intrusion Detection in Distributed Networks. (2026). *Innovative Journal of Computing, Security and Digital Technologies, 1*(1), 15-22.
https://ijcdst.in/index.php/about/article/view/8

Federated learning allows participating clients to train a shared model without directly sending their raw training records to a central server. Each client trains locally and shares model updates, which are combined to produce a global model. However, data locality alone is not a complete privacy guarantee. Model updates can leak information, while compromised clients may submit manipulated updates that degrade detection or introduce backdoors [3].

This paper proposes a conceptual framework for privacy-preserving federated intrusion detection in distributed networks. The framework integrates privacy controls into the training workflow rather than treating federated learning itself as sufficient protection. The main contributions are:

1. A federated intrusion detection workflow for distributed network participants.
2. A privacy layer combining secure aggregation and differential privacy.
3. A discussion of data heterogeneity, poisoning risks, and deployment challenges.
4. An evaluation plan that avoids claiming unmeasured performance.

II. BACKGROUND AND RELATED WORK

A. Intrusion Detection in Distributed Networks

An IDS identifies suspicious activity by analyzing network traffic, system events, or other security signals. Signature-based systems detect known patterns, while machine-learning-based systems can identify statistical or behavioral deviations. In distributed environments, detection models must accommodate differences in network topology, device capabilities, traffic patterns, and attack prevalence.

B. Federated Learning

In a typical federated learning process, a server distributes a model to selected clients. Each client trains the model using its local dataset and returns an update. The server aggregates updates and distributes the resulting global model for another training round. This process can support collaborative learning without centralizing raw records.

A common aggregation method is Federated Averaging (FedAvg), which combines client model parameters according to their local sample counts [4]. In intrusion detection, however, clients may have highly different data distributions. For example, one network may contain mostly benign enterprise traffic, while another may contain a high proportion of IoT traffic or attack records. Such non-independent and identically distributed (non-IID) data can make global training unstable or reduce performance for some participants.

C. Privacy and Security Risks

Federated learning reduces direct sharing of raw data, but it introduces other risks. An observer may infer information from model updates, while a malicious participant may manipulate its update to influence the global model. Reviews of federated intrusion detection emphasize the need to evaluate privacy mechanisms and model robustness rather than assuming that data locality alone provides sufficient protection [1], [3].

III. PROPOSED FRAMEWORK

The proposed framework consists of six connected stages: local data preparation, local model training, privacy protection, secure aggregation, global model distribution, and continuous evaluation.

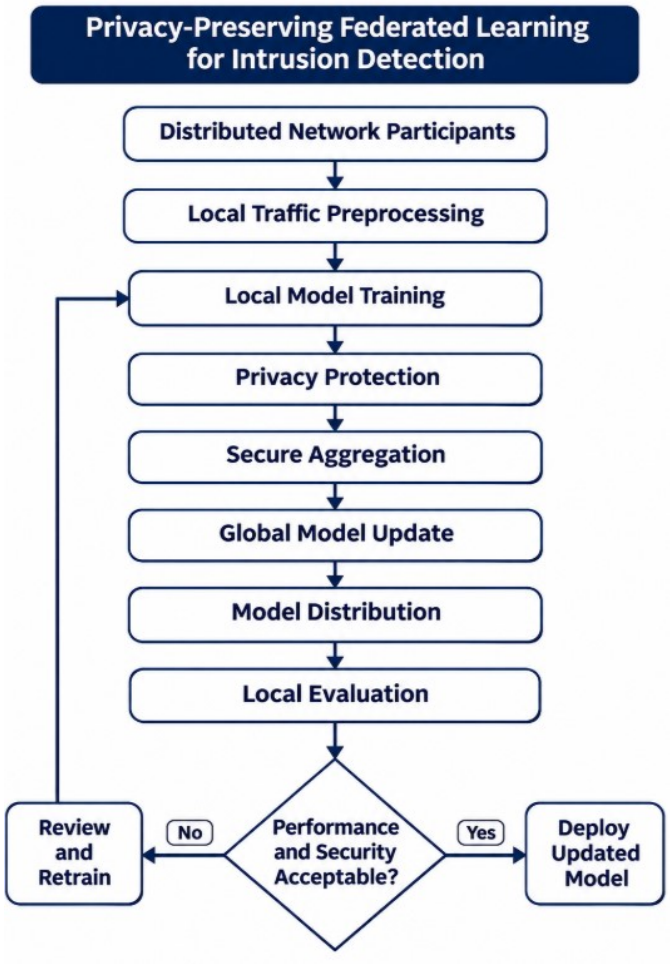


Figure 1. Proposed Privacy-Preserving Federated Learning Workflow for Intrusion Detection

A. Local Data Preparation

Each participant collects and preprocesses its own network data. Preprocessing may include removing invalid records, normalizing numerical features, encoding categorical fields, and assigning labels where available. The feature schema should be consistent across participants so that model updates are compatible.

Raw traffic records remain within the participant's administrative environment. Data minimization should be applied before training, and personally identifying or sensitive fields should be removed or transformed when they are not required for detection.

B. Local Model Training

Each selected participant receives the current global model and trains it using local data. A candidate model may use a multilayer perceptron or another suitable classifier for flow-level features. Model choice should reflect the available compute, memory, and latency constraints.

The local training process produces a model update rather than transferring the underlying traffic records. Participants may differ in data volume, class balance, hardware, and training time. The framework therefore allows partial participation and records client availability and update metadata for evaluation.

C. Privacy Protection

The framework applies privacy mechanisms to reduce the information exposed through model updates.

Differential privacy (DP) adds calibrated noise to updates or training contributions to limit how much an individual record can influence the released model. Its privacy guarantee depends on the clipping method, noise scale, sampling process, and cumulative privacy budget. The privacy parameter should be reported alongside detection performance rather than treated as a hidden implementation detail.

Secure aggregation allows a server to compute an aggregate of client updates without viewing each individual update in plaintext. It can reduce the server's ability to inspect participant-specific contributions. However, secure aggregation introduces protocol and communication overhead and must account for client dropouts.

These mechanisms address different risks: secure aggregation limits direct access to individual updates, while differential privacy bounds the influence of training records on released outputs. Neither mechanism alone protects against every attack.

D. Secure Aggregation and Global Model Update

After local training and privacy processing, clients send protected updates to the aggregation service. The service combines eligible updates and produces a new global model. A basic weighted aggregation rule can be expressed as:

$$w_{t+1} = \sum_{k=1}^{K} \frac{n_k}{\sum_{j=1}^{K} n_j} w_{t+1}^{(k)}$$

This equation describes a basic aggregation approach; it does not by itself provide privacy or robustness. The framework therefore pairs aggregation with update validation and privacy controls. Update validation may check whether updates are malformed, unusually large, or inconsistent with protocol constraints. Such checks should be designed carefully because legitimate participants may naturally produce different updates when their traffic distributions differ.

E. Model Distribution and Local Evaluation

The global model is distributed to participating networks for local evaluation. Each participant measures detection quality using its own validation data and monitors operational factors such as inference latency and resource consumption. The framework supports controlled rollout so that a model can be withheld or rolled back if it fails acceptance criteria.

F. Continuous Monitoring

The system monitors detection performance, client participation, update failures, privacy-budget consumption, and potential poisoning indicators. Changes in traffic patterns may

require retraining or adjustment of the model. Policy and model changes should be versioned so that their effect can be traced during incident investigation.

IV. THREAT MODEL

The framework considers the following threats:

1. Inference from model updates: An attacker attempts to infer sensitive information about local training records from exchanged updates.
2. Malicious client updates: A compromised participant submits manipulated updates to reduce detection effectiveness or introduce a backdoor.
3. Compromised aggregation service: An attacker attempts to inspect individual updates or alter the global model.
4. Data heterogeneity: Differences in traffic distributions cause uneven detection performance across participants.
5. Communication disruption: Unavailable or unreliable clients delay training rounds or produce incomplete participation.

The framework reduces selected risks but does not eliminate them. For example, differential privacy may reduce information leakage while also reducing model utility; secure aggregation may conceal individual updates but complicate the detection of malicious participants. A practical deployment should state which threats it addresses and which remain outside its protection scope.

V. EVALUATION PLAN

The proposed framework requires empirical evaluation before any claims can be made about detection performance or deployment readiness. A future study should use a clearly documented experimental environment with multiple simulated or real network participants.

A. Dataset and Partitioning

A suitable evaluation may use publicly available intrusion detection datasets, such as CICIDS2017 or CSE-CIC-IDS2018, with the chosen version and preprocessing steps documented. Data should be partitioned across clients to represent realistic differences in traffic volume and attack distribution. Both balanced and non-IID partitions should be examined.

B. Baseline Comparisons

The framework should be compared against:

- A centralized model trained using pooled data, where such pooling is permitted for experimental comparison.
- A federated model without additional privacy mechanisms.
- A federated model with secure aggregation.
- A federated model with differential privacy.
- The combined privacy-preserving configuration.

C. Evaluation Metrics

Detection quality should be assessed using precision, recall, F1-score, false-positive rate, and per-class performance. Operational evaluation should include communication volume, training time, client participation rate, inference latency, and resource use. Privacy evaluation should report the mechanisms used and their parameters, including the differential privacy budget where applicable.

Adversarial evaluation should test whether malicious updates can reduce detection performance or introduce targeted errors. Results should include the experimental configuration, client partitioning method, training settings, and limitations. No experimental results are reported in this conceptual article.

VI. CHALLENGES AND LIMITATIONS

A. Non-IID Data

Distributed participants often observe different traffic patterns and attack types. This can cause the global model to favor participants with larger or more common datasets. Personalization, client clustering, or carefully designed aggregation may help, but each approach introduces additional complexity.

B. Privacy–Utility Trade-Off

Stronger privacy protection can reduce the information available for training and may affect detection quality. Differential privacy requires careful calibration, and its cumulative privacy cost should be tracked across training rounds. Privacy settings should be selected according to the threat model and reported transparently.

C. Communication and Resource Constraints

Frequent model updates can consume bandwidth and energy, particularly for edge and IoT participants. Model compression, fewer communication rounds, and partial participation may reduce overhead, but their effects on convergence and detection quality require testing.

D. Poisoning and Robustness

Privacy mechanisms do not automatically prevent malicious updates. A robust design should evaluate update validation, participant authentication, anomaly-aware aggregation, and recovery procedures. These controls must account for legitimate differences between participants so that unusual but valid updates are not incorrectly rejected.

E. Deployment and Governance

Organizations may have different policies for data retention, model sharing, and incident response. A deployment needs clear participant agreements, access controls, model-version tracking, and processes for removing a participant or revoking its credentials.

VII. FUTURE RESEARCH DIRECTIONS

Future work should investigate adaptive privacy budgets, efficient secure aggregation, and defenses against poisoning under non-IID data. Additional research is needed on personalized federated models, cross-domain evaluation, and practical deployment in constrained edge

environments. Studies should report both detection performance and privacy guarantees, rather than presenting data locality as a complete privacy solution.

VIII. CONCLUSION

This paper presented a conceptual privacy-preserving federated learning framework for intrusion detection in distributed networks. The framework combines local training, privacy protection, secure aggregation, global model updates, and continuous evaluation. It recognizes that federated learning can reduce the need to centralize raw traffic data, but does not automatically prevent inference or poisoning attacks. Future empirical work should assess detection quality, privacy guarantees, communication overhead, and robustness under heterogeneous network conditions before the framework is considered ready for operational deployment.

References

1. McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, 1273–1282.
2. Bonawitz, K., et al. (2017). Practical secure aggregation for privacy-preserving machine learning. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 1175–1191. https://doi.org/10.1145/3133956.3133982
3. Abadi, M., et al. (2016). Deep learning with differential privacy. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 308–318. https://doi.org/10.1145/2976749.2978318
4. Geyer, R. C., Klein, T., & Nabi, M. (2017). Differentially private federated learning: A client level perspective. NIPS Workshop on Machine Learning on the Phone and other Consumer Devices.
5. Kairouz, P., et al. (2021). Advances and open problems in federated learning. Foundations and Trends in Machine Learning, 14(1–2), 1–210. https://doi.org/10.1561/2200000083
6. Yin, X., Zhu, Y., & Hu, J. (2021). A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions. ACM Computing Surveys, 54(6), Article 131. https://doi.org/10.1145/3457907
7. Preuveneers, D., et al. (2018). Chained anomaly detection models for federated learning: An intrusion detection case study. Applied Sciences, 8(12), 266 federated?
8. Ferrag, M. A., et al. (2021). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. Journal of Information Security and Applications, 50, 102419. https://doi.org/10.1016/j.jisa.2019.102419
9. Belenguer, A., Pascual, J. A., & Navaridas, J. (2025). A review of federated learning applications in intrusion detection systems. Computer Networks, 258, Article 111023. https://doi.org/10.1016/j.comnet.2024.111023

10. Buyuktanir, B., Altinkaya, Ş., Karatas Baydogmus, G., & Yildiz, K. (2025). Federated learning in intrusion detection: Advancements, applications, and future directions. Cluster Computing, 28, Article 473. https://doi.org/10.1007/s10586-025-05325-w

11. Bunko, T., Johnstone, M. N., Yang, W., et al. (2026). A survey of privacy-preserving federated learning for intrusion detection systems. Artificial Intelligence Review, 59, Article 125. https://doi.org/10.1007/s10462-026-11519-4

12. Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020). Federated learning: Challenges, methods, and future directions. IEEE Signal Processing Magazine, 37(3), 50–60. https://doi.org/10.1109/MSP.2020.2975749