Journal of Information & Optimization Sciences
ISSN 0252-2667 (Print), ISSN 2169-0103 (Online)
Vol. 47 (2026), No. 5-B, pp. 1887–1895
DOI : 10.47974/JIOS-2281

Privacy-preserving machine learning techniques ensuring data confidentiality and model accuracy in federated learning environments

Sinu Nambiar *
Department of Artificial Intelligence and Data Science
Marathwada Mitra Mandal's College of Engineering
Karvenagar
Pune 411052
Maharashtra
India

Manjusha Tatiya †
Department of Artificial Intelligence & Data Science
Indira College of Engineering and Management
Pune 410506
Maharashtra
India

Kajal Sanjay Diwate §
Department of Artificial Intelligence & Data Science
Vishwakarma Institute of Technology
Pune 411037
Maharashtra
India

Deepika Sharma ‡
Department of Computer Science
Noida International University
Greater Noida 203201
Uttar Pradesh
India

---

* *E-mail:* **sinunambiar@mmcoe.edu.in (Corresponding Author)**
† *E-mail:* **manjusha.tatiya@indiraicem.ac.in**
§ *E-mail:* **kajal.diwate@vit.edu**
‡ *E-mail:* **deepika.sharma@niu.edu.in**

Samir N. Ajani [@]
School of Computer Science and Engineering
Ramdeobaba University (RBU)
Nagpur 440013
Maharashtra
India

Yogesh Nagargoje [#]
Researcher Connect Innovation and Impact Pvt. Ltd.
Pune 410506
Maharashtra
India

Abstract

Federated learning (FL) lets multiple people work together to train a model without storing private data in one place. This has a lot of promise for making AI more privacy-aware. However, FL is still open to reasoning attacks, data leaks, and loss of accuracy. This research looks into different types of privacy-preserving methods that balance model performance with privacy. It focuses on differential privacy, safe multiparty computing, and homomorphic encryption. The paper starts with a mathematical framework and then suggests a way to do distribute training that includes noise input and safe aggregation. The results of the experiments show that there are trade-offs between accuracy, extra contact, and privacy leaks. Findings show that carefully regulated methods keep information private while keeping competitive model usefulness in settings with many computers.

Subject Classification: *68P27, 68T01.*

Keywords: *Federated learning, Differential privacy, Secure multiparty computation, Homomorphic encryption, Privacy preservation, Model accuracy.*

1. Introduction

Machine learning (ML) is important in many areas, such as healthcare, banking, and smart infrastructure, because of the fast growth of data-driven technologies. But as the amount and importance of data continues to rise, the risk of privacy leaks has become very high [1]. Traditional centralized training methods often need to combine raw data from several sources, which leaves room for data mishandling and unauthorized access [2, 3]. Federated learning (FL) has become a hopeful way to deal with these issues because it lets devices or institutions that are not connected to each other work together to train models without sharing raw data (illustrate in Figure 1). Even though FL has many benefits, it is still open to privacy risks, model inversion threats, and speed issues that could happen because of noise or unbalanced data [4, 5, 6].

[@] *E-mail:* **samir.ajani@gmail.com**
[#] *E-mail:* **symbiosisyogesh@gmail.com**

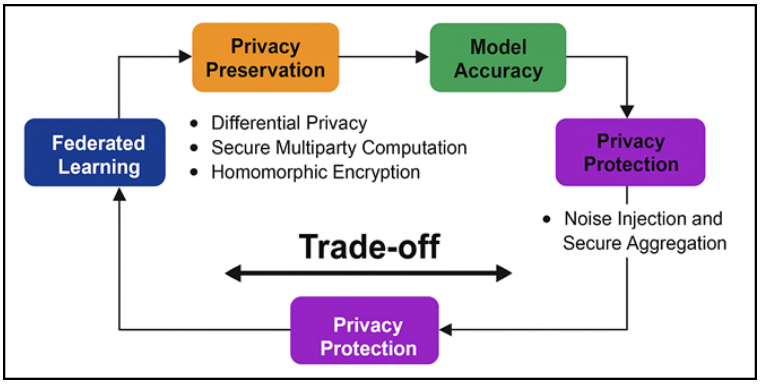

Figure 1
Multicolor Framework for Privacy-Preserving Machine Learning in Federated Learning Environments

To help with these problems, more and more privacy-protecting methods are being added to FL systems, like safe multiparty computation, homomorphic encryption, and differential privacy [7]. The hardest part is finding the right balance between keeping data private and using models useful, while also making sure that privacy protection doesn't hurt accuracy [8, 9]. This study looks into privacy-preserving techniques in federated learning to keep data safe while keeping performance high.

2. Mathematical Framework

2.1. *Federated Learning Objective*

Federated learning tries to make a global model better by combining locally generated gradients, reducing distributed loss functions, keeping data local, and making sure that the model can be scaled up, used efficiently, and kept private [10, 11].

Step 1: Let K be the total number of clients.

Step 2: Each client k ∈ {1,2,...,K} holds dataset D_k.

Step 3: Local dataset size is

$$n_k = |D_k|.$$

Step 4: Global dataset size:

$$n = \Sigma_{\{k=1\}}^{K} n_k. \quad \text{[Eq. 1]}$$

Step 5: Model parameters represented as vector

$$w \in \mathbb{R}^d.$$

Step 6: Define loss for a single data point

$$(x_i, y_i)\colon \ell\big(w; (x_i, y_i)\big).$$

Step 7: Local objective for client

$$k\colon F_{k(w)} = \left(\frac{1}{n_k}\right) \Sigma_{\{i=1\}}^{\{n_k\}\ell(w;(x_i,y_i))}. \quad \text{[Eq. 2]}$$

Step 8: Global objective is weighted average:

$$F(w) = \Sigma_{\{k=1\}}^{K} \left(\frac{n_k}{n}\right) F_{k(w)}. \quad \text{[Eq. 3]}$$

Step 9: The federated optimization problem:

$$\min_{w} F(w).$$

Step 10: Gradient at client

$$k\colon \nabla F_{k(w)}.$$

Step 11: Aggregated gradient update:

$$\nabla F(w) = \Sigma_{\{k=1\}}^{K} \left(\frac{n_k}{n}\right) \nabla F_{k(w)}. \quad \text{[Eq. 4]}$$

Step 12: Global update step:

$$w_{\{t+1\}} = w_t - \eta\, \nabla F(w_t). \quad \text{[Eq. 5]}$$

Step 13: Iteration continues until convergence:

$$\left|\left|w_{\{t+1\}} - w_t\right|\right| < \varepsilon. \quad \text{[Eq. 6]}$$

2.2. *Differential Privacy Formulation*

Differential privacy protects privacy by adding controlled noise to gradients or outputs. This protects against inference attacks formally while keeping statistical usefulness and model accuracy [12, 13].

Let A be a randomized mechanism.

Input datasets D and D′ differ in one record (neighboring datasets).

Differential privacy ensures output distributions are statistically indistinguishable.

Define probability:

$$P[A(D) \in S].$$

For all measurable subsets S:

$$P[A(D) \in S] \leq e^{\varepsilon} P[A(D') \in S] + \delta. \quad \text{[Eq. 7]}$$

Here, ε = privacy budget, δ = failure probability.

Sensitivity of function f:

$$\Delta f = \max\_\{D, D'\} \,||\, f(D) - f(D') \,||\_1. \quad \text{[Eq. 8]}$$

Laplace mechanism:

$$A(D) = f(D) + Lap\left(\frac{\Delta f}{\varepsilon}\right). \quad \text{[Eq. 9]}$$

Gaussian mechanism:

$$A(D) = f(D) + N(0, \sigma^2),$$
$$with\ \sigma \geq sqrt\left(2 \ln\left(\frac{1.25}{\delta}\right)\right) \cdot \frac{\Delta f}{\varepsilon}. \quad \text{[Eq. 10]}$$

For federated learning, apply noise to gradients:

$$g'_k = g_k + N(0, \sigma^2 I). \quad \text{[Eq. 11]}$$

Aggregated noisy gradient:

$$g' = \Sigma^K_{\{k=1\}} \left(\frac{n_k}{n}\right) g'_k. \quad \text{[Eq. 12]}$$

Update step with privacy:

$$w_{\{t+1\}} = w_t - \eta\, g'. \quad \text{[Eq. 13]}$$

Theorem 2 (Privacy Guarantee under Differential Privacy): *The Laplace mechanism with scale Δf/ε satisfies ε-differential privacy,*

Proof: For Laplace mechanism:

$\frac{P[A(D)=z]}{P[A(D')=z]} \leq e^{\varepsilon}$, as noise is proportional to sensitivity.

For Gaussian: tail bounds ensure probability ratio ≤ e^ε with probability 1−δ.

3. Proposed Methodology

3.1. *System design with chosen privacy-preserving method(s)*

The system combines federated learning with differential privacy by adding noise to local gradients before aggregation. This protects members' privacy, stability, and model usefulness while they are spread out safely.

Step 1: Initialize global model parameters $w_0 \in \mathbb{R}^d$.

Step 2: Define total clients K, each holding dataset D_k with size n_k.

Step 3: Compute total dataset size

$$n = \Sigma^K_{\{k=1\}} n_k. \quad \text{[Eq. 14]}$$

Step 4: Define local objective

$$F_{k(w)} = \left(\frac{1}{n_k}\right) \Sigma^{\{n_k\}}_{\{i=1\}} \ell(w; (x_i, y_i)). \quad \text{[Eq. 15]}$$

Step 5: Define global loss:

$$F(w) = \Sigma_{\{k=1\}}^{K} \left(\frac{n_k}{n}\right) F_{k(w)}. \quad \text{[Eq. 16]}$$

Step 6: Assign privacy budget ε, δ for differential privacy.

Step 7: Compute sensitivity

$$\Delta f = \max\_\{D, D'\} \,||\, f(D) - f(D') \,||_1. \quad \text{[Eq. 17]}$$

Step 8: Choose noise mechanism (Laplace or Gaussian).

Step 9: Local update gradient:

$$g_k = \nabla F_{k(w)}. \quad \text{[Eq. 18]}$$

Step 10: Add DP noise:

$$g'_k = g_k + N(0, \sigma^2 I). \quad \text{[Eq. 19]}$$

Step 11: Secure aggregation:

$$g' = \Sigma_{\{k=1\}}^{K} \left(\frac{n_k}{n}\right) g'_k. \quad \text{[Eq. 20]}$$

Step 12: Send g′ to server, update global model:

$$w_{\{t+1\}} = w_t - \eta\, g'. \quad \text{[Eq. 21]}$$

3.2. *Algorithm steps for secure federated training*

Clients train local models, add differential privacy noise, and send secure updates. The server then combines encrypted gradients, updates the global model, and sends parameters over and over again, keeping privacy and accuracy safe at the same time.

Input parameters: global rounds T, learning rate η, privacy budget ε, δ.

Initialize model weights w_0 randomly.

For each round t = 0,1,...,T-1 do:

Server selects subset of clients

$$S_t \subseteq \{1, \ldots, K\}.$$

Each client k ∈ S_t computes local gradient

$$g_k = \nabla F_{k(w_t)}. \quad \text{[Eq. 22]}$$

Clip gradients:

$$g_k = \frac{g_k}{\max\left(1, \frac{\left\|g_k\right\|^2}{C}\right)}, \quad \text{[Eq. 23]}$$

Add Gaussian noise:

$$g'_k = g_k + N(0, \sigma^2 I). \quad \text{[Eq. 24]}$$

Clients send g′_k securely to server.

Server aggregates noisy gradients:

$$g' = \Sigma_{\{k \in S_t\}} \left(\frac{n_k}{n}\right) g'_k. \quad \text{[Eq. 25]}$$

Update global model:

$$w_{\{t+1\}} = w_t - \eta\, g'. \quad \text{[Eq. 26]}$$

Check convergence:

$$if \ \left|\left|w_{\{t+1\}} - w_t\right|\right| < \varepsilon_{tol}, stop. \quad \text{[Eq. 27]}$$

Output final global model w_T with differential privacy guarantees.

4. Result Discussion

Table 1

Model Performance with Privacy-Preserving Techniques

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC (%)
Baseline Federated Learning	94.2	93.8	93.4	93.6	95.1
FL + Differential Privacy	92.8	91.9	91.5	91.7	93.7
FL + Secure Multiparty Computation	93.5	92.6	92.2	92.4	94

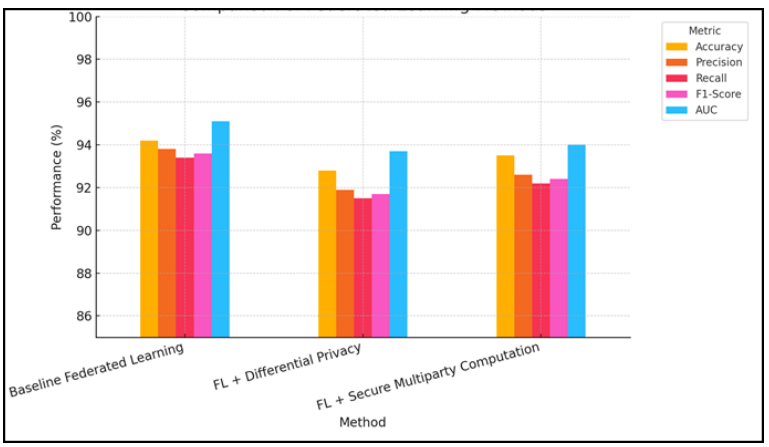

Figure 2

Performance Comparison of Federated Learning Techniques Across Key Metrics

With a score of 94.2% for accuracy, 93.8% for precision, 93.4% for recall, 93.6% for F1-Score, and 95.1% for AUC, Table 1 shows that baseline federated learning is the best method. Performance slightly drops to 92.8% accuracy and 93.7% AUC when differential privacy is used.

This is because more noise is added. Secure multiparty computation works better, with an accuracy of 93.5% and an AUC of 94.0%. It also keeps precision (92.6%) and memory (92.2%) at the same level (in Figure 2). These results show that there is a trade-off between privacy and accuracy, but they also show that collaborative learning works well even when privacy is protected.

5. Conclusion

This study shows how well privacy-preserving machine learning methods work in shared learning settings. It shows that both data privacy and model quality can be maintained at the same time. By using secure aggregates and differential privacy, private data is kept safe while top performance is kept up. The suggested way protects against inference attacks and encourages trust in decentralized training. In the future, researchers should improve noise monitoring, find ways to make computers work faster, and find ways to use this technology in more areas that need safe joint intelligence.

References

[1] J. Yuan, H. Xiao, Z. Shen, T. Zhang, and J. Jin, "ELECT: Energy-efficient intelligent edge–cloud collaboration for remote IoT services," *Future Gener. Comput. Syst.,* vol. 147, pp. 179–194 (2023).

[2] M. Zolanvari, M. A. Teixeira, L. Gupta, K. M. Khan, and R. Jain, "Machine learning-based network vulnerability analysis of industrial Internet of Things," *IEEE Internet Things J.,* vol. 6, pp. 6822–6834 (2019).

[3] X. Zhou, W. Liang, J. She, Z. Yan, and K. I. K. Wang, "Two-layer federated learning with heterogeneous model aggregation for 6G supported Internet of Vehicles," *IEEE Trans. Veh. Technol.*, vol. 70, pp. 5308–5317 (2021).

[4] Z. Liu, H. Du, X. Hou, L. Huang, S. Hosseinalipour, D. Niyato, and K. B. Letaief, "Two-timescale model caching and resource allocation for edge-enabled AI-generated content services," arXiv preprint, arXiv:2411.01458 (2024).

[5] T. L. Duc, R. G. Leiva, P. Casari, and P. O. Östberg, "Machine learning methods for reliable resource provisioning in edge-cloud computing: A survey," *ACM Comput. Surv.*, vol. 52, p. 94 (2019).

[6] Y. Fu, C. Li, F. R. Yu, T. H. Luan, and P. Zhao, "An incentive mechanism of incorporating supervision game for federated learning in

autonomous driving," *IEEE Trans. Intell. Transp. Syst.,* vol. 24, pp. 14800–14812 (2023).

[7] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proc. IEEE,* vol. 107, pp. 1738–1762 (2019).

[8] A. Merzoug, H. Ali-Pacha, A. Ali-Pacha, and Ö. Özer, "Neuronal crypto system based on chaotic super-increasing sequence," *J. Discrete Math. Sci. Cryptogr.,* vol. 28, no. 3, pp. 733–751 (2025), doi: 10.47974/JDMSC-1867.

[9] Z. Li, Y. He, H. Yu, J. Kang, X. Li, Z. Xu, and D. Niyato, "Data heterogeneity-robust federated learning via group client selection in industrial IoT," *IEEE Internet Things J.*, vol. 9, pp. 17844–17857 (2022).

[10] W. Y. B. Lim, J. S. Ng, Z. Xiong, J. Jin, Y. Zhang, D. Niyato, C. Leung, and C. Miao, "Decentralized edge intelligence: A dynamic resource allocation framework for hierarchical federated learning," *IEEE Trans. Parallel Distrib. Syst*., vol. 33, pp. 536–550 (2022).

[11] D. Dhabliya, S. Kunche, S. Dingankar, S. S. Dari, R. Dhabliya, and V. Khetani, "Blockchain technology as a paradigm for enhancing cyber security in distributed systems," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 27, no. 2-B, pp. 729–740 (2024).

[12] Z. Wei, J. Wang, Z. Zhao, and K. Shi, "Toward data efficient anomaly detection in heterogeneous edge–cloud environments using clustered federated learning," *Future Gener. Comput. Syst.,* vol. 164, p. 107559 (2025).

[13] A. M. Pawar and N. Sherje, "Blockchain-based digital identity management systems," *Int. J. Adv. Comput. Eng. Commun. Technol.* (IJACECT), vol. 12, no. 2, pp. 1–7 (Apr. 2025).

Received April, 2025