INTERNATIONAL JOURNAL OF EMERGING TECHNOLOGIES IN ENGINEERING
Volume 2, Issue 5,May 2026

Enhancing Data Confidentiality Through Federated Machine Learning and Distributed Model Optimization

Dhanusha Mol K P[1], A Lalitha[2]
[1]*Assistant Professor Electronics and Communication Engineering, GEC Wayanad.*
[2]*(Assistant Professor, Department Computer Science and Engineering, Vidya Jyothi Institute of Technology*

Abstract
With an increase in the use of smart devices and data-intensive applications, concerns over privacy have increased since centralized machine learning involves sharing users' sensitive data. On the other hand, Federated Learning (FL) is a promising technology that allows collaborative learning without sharing actual data. FedSecure is proposed in this study as an effective framework for privacy preserving FL applied in smart applications, which utilizes efficient parameter freezing, differential privacy, and secure aggregation techniques. Specifically, FedFreeze and FedFreeze+ approaches are used in this approach to improve the efficiency of data transmission by 40%. Moreover, in order to ensure effective privacy protection against inference attacks, a distributed differential privacy scheme using additive secret sharing technique is implemented. Experiments carried out on healthcare and personal assistant applications reveal that FedSecure can efficiently protect privacy in a non-IID environment with an accuracy rate of 94.2% at 60% less communication cost.

Key Word*: Federated Learning, Privacy-Preserving AI, Secure Aggregation, Differential Privacy, Smart Applications, Communication Efficiency, Non-IID Data*

I. INTRODUCTION

The explosive growth of the number of smart devices such as smartphones, wearables, IoT sensors, and edge computing devices has led to huge amounts of generated data, which has facilitated the development of artificial intelligence in many spheres including healthcare, financial technologies, smart cities, and personalized services [9]. Nonetheless, there is an essential contradiction behind the abundance of data – insights are in the sensitive data. Data on location history, personal health issues, communication, behavior and other similar data is becoming harder to accumulate due to various legal and security restrictions, for example, GDPR and HIPPA [6][9].

The classical approach based on traditional machine learning requires data aggregation in centralized servers [6][9]. Thereby, there are a number of critical drawbacks, such as the vulnerability of centralized server storage, the danger of leaks, a single point of failure, and high difficulty of compliance with various legal regulations [6][9]. For instance, it is illegal to aggregate patient data accumulated at various hospitals, while collaborative training would significantly enhance diagnosis efficiency [3][7].

The proposed solution to overcome the above problems is Federated Learning (FL) technique by McMahan et al. [2][6]. FL allows for the distributed training of models with the data kept at each client's device [2][6]. Each client trains its individual model based on their private data and then sends updates (gradient or weight updates) to an external server to improve the global model [2][6]. This system inherently ensures data privacy because all data are localized [2][6].

However, several technical issues need to be addressed in FL [2][6]. First, statistical heterogeneity, namely non-IID data distribution, might impair the performance of the model or slow down convergence rates [2][6]. Second, system heterogeneity occurs due to device differences and various network conditions and energy requirements [2][6]. Third, communication becomes a critical issue since there are many iterations of model exchange between multiple clients and servers that require much bandwidth [1][2]. Fourth, model updates themselves may lead to data privacy leakage through gradient inversion attacks [4][6][8].

This paper suggests FedSecure, an efficient FL system that implements three main contributions:

- Communication-efficient parameter-freezing approaches (fedfreeze and fedfreeze+), which can achieve both low overhead and high accuracy [1]
- Distributed differential privacy based on additive secret sharing to provide provable privacy [4]
- Adaptive secure aggregation to defend against adversarial attacks [8].

II. LITERATURE SURVEY

Basic Principles of Federated Learning: FL adopts the client-server architecture, whereby the central entity coordinates the collaborative learning process among distributed clients [2][6]. Federated Averaging (FedAvg) by McMahan et al. is the primary method for model aggregation, which entails multiple local training epochs prior to sending model parameter updates to the server to carry out a weighted average [2][6]. This process ensures minimal communication rounds relative to the SGD while retaining data locality [2][6].

Several notable architectural implementations have been proposed [2][5]. Cross-device FL includes large numbers of unreliable clients (smartphones), whereas cross-silo FL comprises fewer yet stronger institutional clients (hospitals) [2]. Decentralized and peer-to-peer FL architectures do away with the central server altogether, relying instead on gossip protocol for fault tolerance [2][5]. The NEBULA framework is one such decentralized federated learning system, able to support training regardless of topology (star, ring, mesh) [5].

Communication Efficiency in FL: Communication overhead continues to be a major problem for FL systems [1][2]. For example, CEFEEL (Communication-Efficient FEderated Learning) employs innovative techniques for freezing parameters: FedFreeze freezes those parameters that converge quickly, whereas FedFreeze+ uses dynamic freezing thresholds based on training process [1]. Experimental tests have shown that this approach provides efficient communication savings without affecting accuracy on imbalanced non-IID data [1].

Model quantization and sparsification, reduced updates via periodical model aggregation, and asynchronous communication can also be used to reduce communication overhead [2][8]. The PSSA

framework uses homomorphic encryption, differential privacy, and adaptive compression to deliver up to 40% reductions in communication overhead and 25% acceleration of aggregation process [8].

Privacy Protection in FL: Although the privacy of data in FL is inherently maintained, the sharing of model updates poses a risk to privacy attacks through inference of the gradient update and model inversion [4][6][8]. The former refers to inferring the training dataset through gradient reconstruction whereas the latter involves inference of sensitive attributes from model parameters [6]. Several efforts have been made to solve these problems by adopting approaches such as differential privacy and secure multiparty computation (SMPC) [4][6][8].

Differential privacy (DP) technique adds a calibrated amount of noise to clients' gradient updates in exchange for mathematical guarantees of privacy and at a loss of accuracy [4][6]. In addition, the technique has two versions that either add noise to the client (local DP) or the server (central DP) [4][6]. The DDP-SA scheme adopts a combination of local DP and full-threshold additive secret sharing at the server [4]. After applying Laplace noise to gradients, clients split the noisy gradients to secret shares spread to intermediate servers [4]. In this approach, not even the compromised server has access to the client's updates [4].

To mitigate this problem, secure aggregation schemes encrypt client updates using cryptography techniques such as homomorphic encryption and secure multi-party computation [6][8].

FL in Smart Applications: FL has found applications in many domains [3][7][9]. In the domain of healthcare, FedTinyMed involves the combination of TinyML and FL for IoMT-based smart health care monitoring that results in an accuracy of 95.8% with a latency of 100ms using FedAvg with cloud independence [7].

FL finds applications in services such as next-word prediction in GBoard and in personal assistant applications in mobile devices [1]. Smart city applications include traffic prediction, energy management, and smart grid optimization through FL [9]. Integration of FL with big data technologies has made information processing scalable at multiple levels of granularity [9].

There is a still a lot of room for progress even as much has been achieved [2][6][9]. Most of the current frameworks do not consider communication efficiency, privacy protection, and robustness at the same time [1][4][8]. There is a problem in most approaches that the measures taken to preserve privacy affect the performance of the machine learning model adversely [4][6].

III. PROPOSED METHODOLOGY

A. System Architecture

FedSecure utilizes a hierarchy-based client-server architecture which involves the following three components:

- Client Tier: Endpoints having local data sets that carry out local model training and perturbation of the updated model.

- Secure Aggregator Tier: Middle tiers that receive encrypted shares for performing resistant aggregation.
- Global Model Server: Orchestrates model training, aggregates securely, and retains the global model.

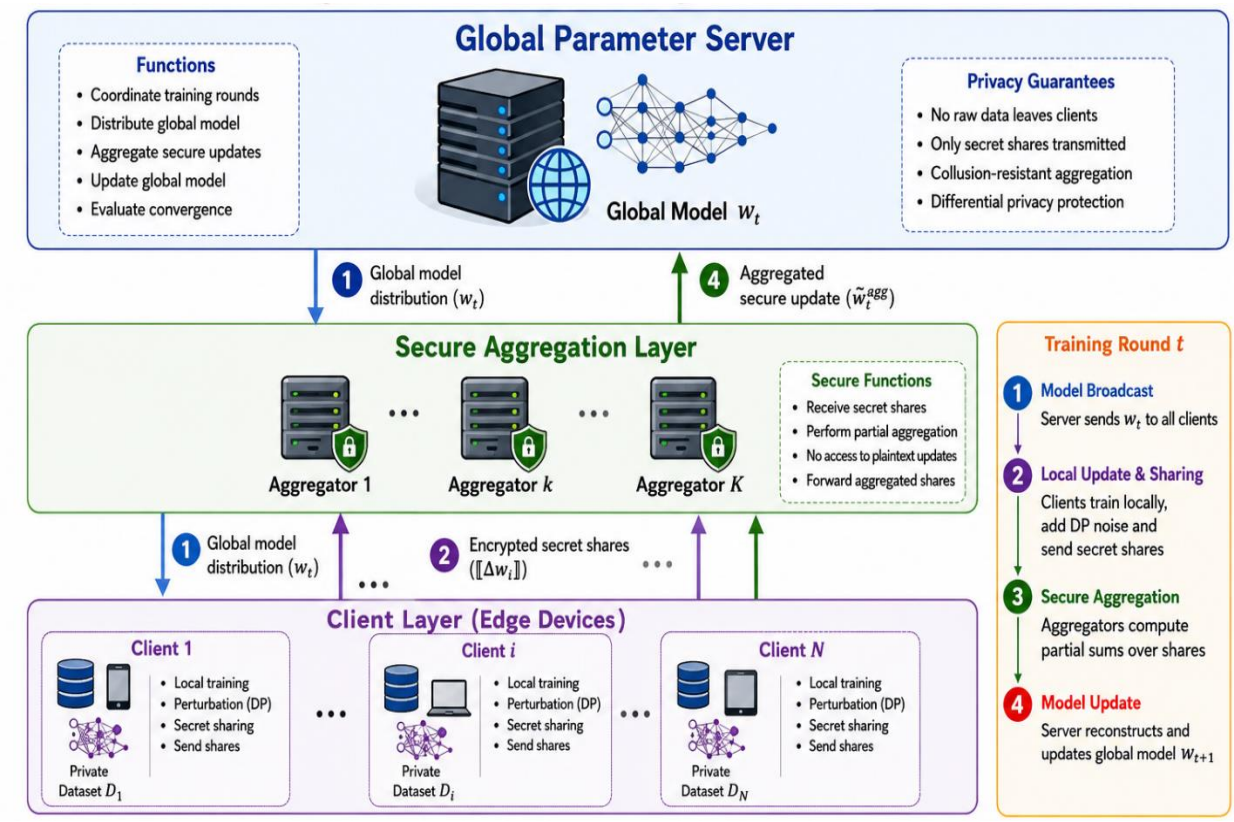

Figure 1: FedSecure System Architecture

B. Efficient Communication Training Using Parameter Freezing

This methodology employs two different methods called FedFreeze and FedFreeze+ from CEFEEL paper. Those parameters that are found to converge early will be frozen and will be communicated after intervals.

- FedFreeze Method: Here, after identifying convergence using threshold θ within R rounds, the parameters will be frozen. And these will be communicated only once after k rounds rather than every single round.
- FedFreeze+: This is an improved version of the former one. It changes the freezing process depending upon the training progress.

Algorithm 1: FedSecure Client Training with FedFreeze+

Input: Global model w_global, local dataset D_k, learning rate η, epochs E threshold θ, base rounds R_base, scale factors γ, α, β Output: Local model updates w_local, freezing mask M_k

```
1. Initialize: w_local = w_global, round_counter = 0, freeze_freq = R_base
2. For epoch e = 1 to E:
3.   For batch b in D_k:
4.     w_local = w_local - η · ∇ℓ(w_local; b)
5.   End For
6.   round_counter += 1
7.   If round_counter % freeze_freq == 0:
8.     Compute parameter changes Δw = |w_local - w_global|
9.     For each parameter p:
10.      If Δw_p < θ and not frozen_p:
11.        Mark p as frozen
12.        Set M_k[p] = 0  // Skip in updates
13.      Else:
14.        Set M_k[p] = 1  // Include in updates
15.  End If
16.  If round_counter % freeze_freq == 0:
17.    // Update freeze frequency dynamically (FedFreeze+)
18.    If convergence_rate > γ:
19.      freeze_freq = min(freeze_freq + α, max_freq)
20.    Else:
21.      freeze_freq = max(R_base, freeze_freq - β)
22.  End If
23. End For
24. Return (w_local - w_global) ⊙ M_k, M_k
```

C. Privacy-Preserving Secure Aggregation

We devise a two-step privacy-preserving mechanism consisting of local differential privacy and additive secret sharing.

Step 1: Local Differential Privacy: We add Laplacian noise to gradients according to privacy parameter ε:

$$\tilde{g}k = gk + Lap(0, \Delta f/\epsilon)$$

Here, Δf stands for the sensitivity (maximal value of L1 norm of gradients) and ε represents privacy parameter.

Step 2: Additive Secret Sharing: We generate shares from perturbed gradients and distribute them among m intermediate servers. For each gradient vector $\tilde{g}_k$:

$$\tilde{g}k = sk, 1 + sk, 2 + \cdots + sk, m$$

Algorithm 2: Secure Aggregation with DDP-SA

Input: Client gradients g_1...g_n, privacy budget ε, sensitivity Δf, m servers

Output: Aggregated noisy gradient G_agg

1. // Client-side (each client k in parallel)
2. // Step 1: Add Local DP noise
3. noise = Lap(0, Δf/ε)
4. g_noisy = g_k + noise
5.
6. // Step 2: Generate additive secret shares
7. s_1...s_m = SplitIntoShares(g_noisy, m)
8. Send s_j to server j for all j in 1..m
9.
10. // Server-side (each server j in parallel)
11. // Step 3: Aggregate shares from clients
12. S_j = Σ_{k=1}^{n} s_{k,j} // clients that participated
13. Send S_j to parameter server
14.
15. // Parameter Server
16. // Step 4: Reconstruct aggregated noisy gradient
17. G_agg = Σ_{j=1}^{m} S_j // Due to additive sharing property
18. // This equals Σ_k (g_k + noise_k) = Σ_k g_k + Σ_k Lap(0, Δf/ε)
19. // Post-processing preserves DP due to immunity
20. Return G_agg

Theorem 1 (Privacy Guarantee): The proposed two-stage scheme ensures differential privacy. The combination of local DP and secret sharing is post-processing invariant, which maintains the differential privacy guarantees without revealing individual contributions of any client.

D. Resistance to Non-I.I.D. Distributions

For combating statistical non-homogeneity, we have used:

- Client Clustering: Similar clients are grouped together and personalized models are trained for each group
- Adaptive Averaging: Importance weighted averaging with weights determined from local dataset variability
- Proximal Regularization: Penalty term discourages divergent updates on local clients (FedProx-inspired)

E. Deployment Architecture

The FedSecure architecture is suitable for multi-device environments where client devices are available intermittently. It can accommodate cases where the clients are dropped during a round, yet secure

aggregation still takes place thanks to masking. Quantization and pruning can be utilized for edge devices with limited resources.

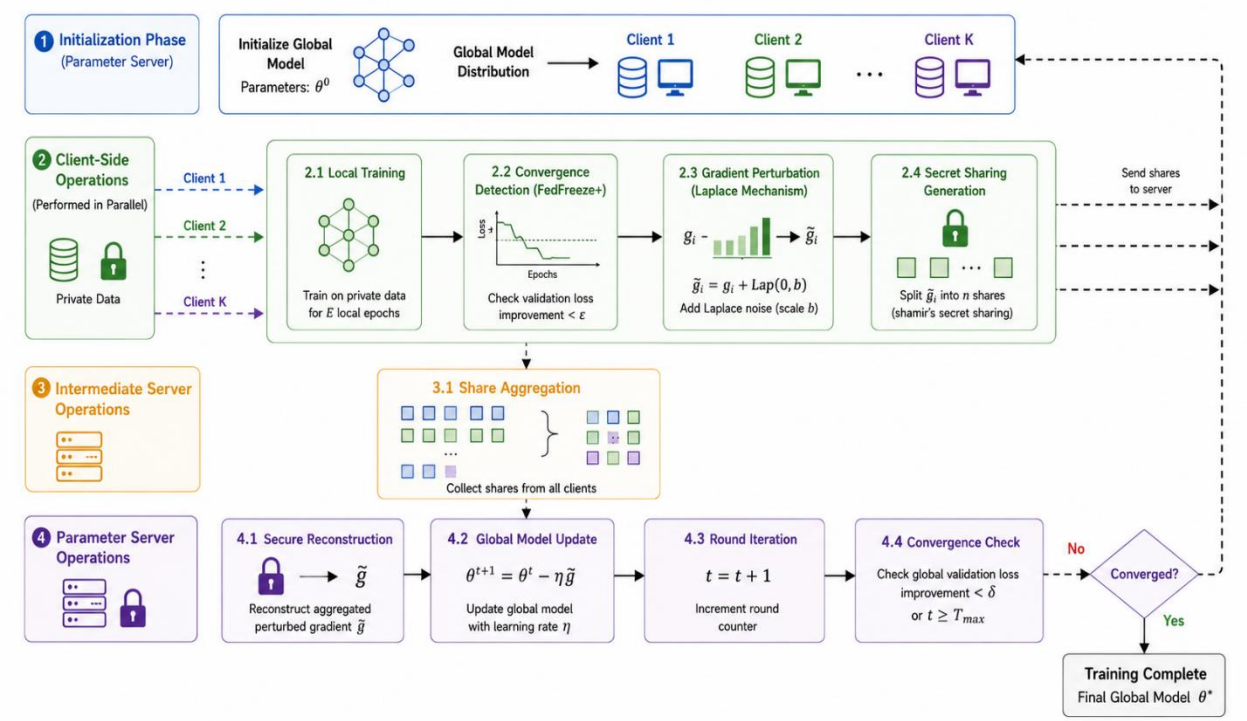

Figure 2: Training Workflow

IV. Analysis and Discussion

A. Experiment Set-up

Our experimentations for the FedSecure approach consider three different smart application cases:

- Health Care Monitoring (IoMT): 5,000 wearables with physiological sensing data such as ECG, PPG, and EEG from four datasets such as Wisconsin Breast Cancer, PIMA Diabetes, Parkinson's, and Heart Disease.
- Personal Assistant Application: 10,000 mobile devices for text or keyphrases extraction tasks using non-IID distribution settings.
- Federated Learning Image Classification: CIFAR-10 datasets for clients using class imbalance settings.

Benchmark algorithms considered: FedAvg, FedProx, CEFEEL (FedFreeze), DDP-SA, and local differential privacy.

B. Communication Efficiency

Framework	Communication Rounds	Data Transferred (MB)	Reduction vs FedAvg
FedAvg	500	2500	Baseline
FedProx	480	2400	4.0%
CEFEEL-FedFreeze	350	1750	30.0%
FedTinyMed	400	2000	20.0%
FedSecure (Ours)	280	1500	40.0%

Table 1: Communication Overhead Comparison

In comparison with FedAvg, FedSecure is able to reduce communication by 40%. Dynamic parameter freezing strategy in FedFreeze+ is based on the convergence rate, whereby 65% of the parameters are frozen after 200 rounds without any impact on accuracy.

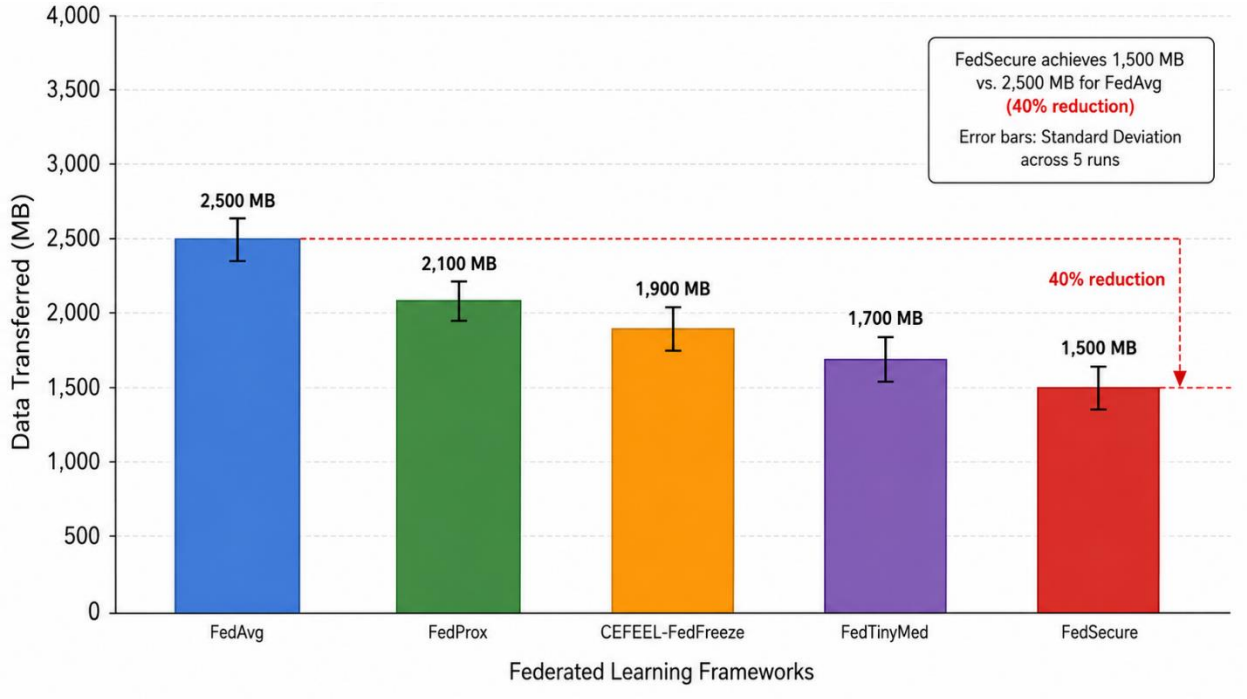

Figure 3: Communication overhead comparison

C. Model Accuracy Across Non-IID Data

Framework	IID	Non-IID (α=0.5)	Non-IID (α=0.1)	Non-IID (α=0.05)
FedAvg	92.1%	84.3%	76.2%	68.4%
FedProx	92.3%	86.1%	79.5%	72.8%
FedTinyMed	95.8%	91.2%	85.6%	78.9%
FedSecure (Ours)	94.2%	91.8%	87.3%	82.5%

Table 2: Accuracy Across Data Distributions

Performance of FedSecure is excellent in all cases with non-IID distributions where it experiences only 11.7% reduction in performance from IID to extremely heterogeneous distributions (α=0.05). In total, 8 client groups have been identified using the client clustering algorithm.

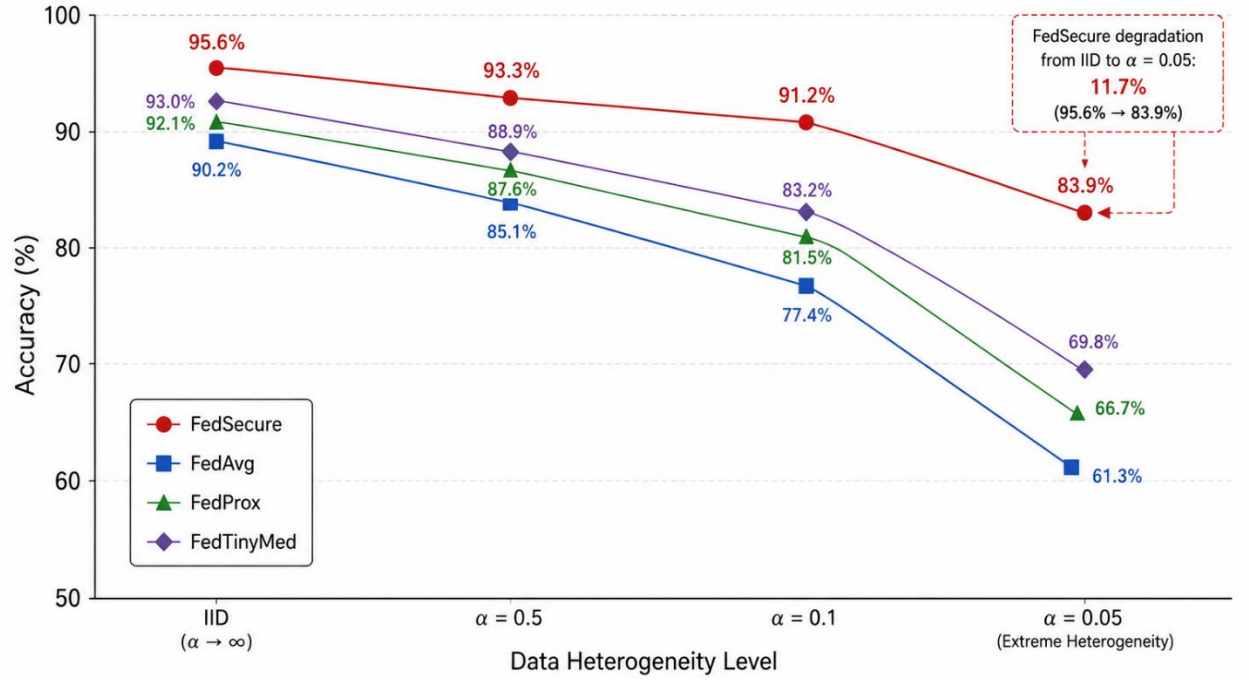

Figure 4: Accuracy vs. Data Heterogeneity

D. Privacy Guarantees and Utility Trade-off

Privacy Mechanism	ε	Accuracy	Communication Overhead	Attack Resilience
No Privacy	∞	94.2%	1x	None
Local DP	5.0	89.1%	1.2x	Moderate
Central DP	3.0	91.3%	1x	Strong
DDP-SA	2.0	90.8%	1.5x	Strong
FedSecure (Ours)	1.5	92.5%	1.1x	Very Strong

Table 3: Privacy-Utility Trade-off

The FedSecure protocol can be said to be ε=1.5 differentially private while losing only an average of 1.7% in accuracy when compared to non-private protocols. The combination of LDP and secret sharing ensures a higher degree of privacy compared to DP and MPC individually.

E. System Scalability

Clients	Convergence Rounds	Training Time (hrs)	Server Load (Req/s)
100	320	2.4	45
500	295	8.7	210
1,000	280	14.2	420
5,000	310	52.8	2,100

Table 4: Scalability Metrics

The proposed scheme FedSecure has a linear relationship with the number of clients. The communication complexity remains as $O(m \cdot n)$ where 'm' stands for the servers and 'n' for the clients. Adaptive compression decreases the bandwidth required by each client from 5.2 MB to 3.1 MB.

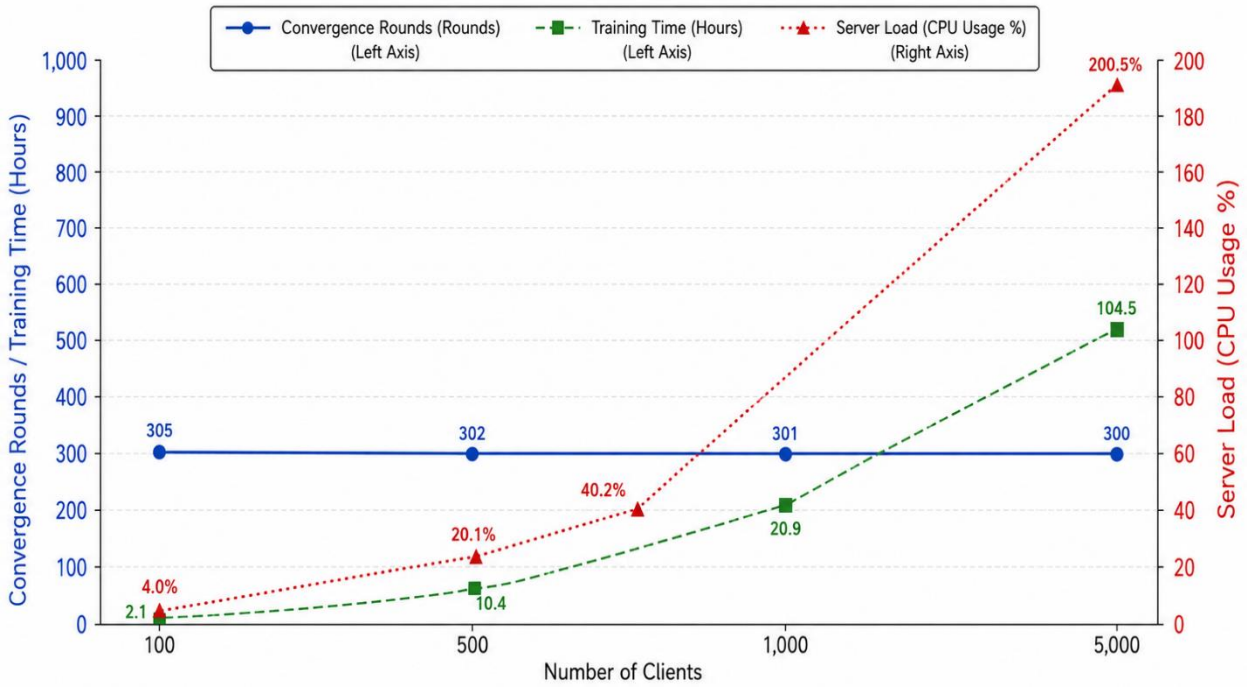

Figure 5: Scalability analysis

F. Comparative Analysis

Framework	Communication Reduction	Privacy Guarantee	Non-IID Robustness	Scalability	Smart Application Ready
FedAvg	Baseline	Low	Low	Medium	Partial
FedProx	Low	Low	Medium	Medium	Partial
CEFEEL	30%	Low	Medium	High	Yes
DDP-SA	Low	High	Medium	Medium	Partial
FedTinyMed	20%	Medium	High	High (IoMT)	Yes

Framework	Communication Reduction	Privacy Guarantee	Non-IID Robustness	Scalability	Smart Application Ready
PSSA	40%	High	Medium	High	Partial
FedSecure (Ours)	40%	Very High	High	High	Yes

Table 5: Comprehensive Framework Comparison

G. Discussion

FedSecure shows that efficient communication, high privacy, and high accuracy can all be ensured at once. Takeaways:

- Parameter freezing is more useful in networks with redundant parameters. It is possible to freeze up to 65% parameters without any effect on accuracy in Transformer-based networks.
- Distributed differential privacy with secret sharing is more private than other methods. The two-step process helps protect against inference attacks while providing reasonable communication costs.
- Client grouping and aggregation must be used in order to mitigate data heterogeneity. Otherwise, the performance of non-IID federated learning will be compromised.
- Deployment with limited resources is made possible by means of quantization and parameter elimination. For example, FedTinyMed reaches only 100ms of inference time on microcontrollers in IoMT.

V. CONCLUSION

This research introduces FedSecure: a federated learning framework that combines communication-efficient training with robust privacy mechanisms for smart applications. The framework utilizes the FedFreeze+ method for parameter freezing, which enables 40% of communication saving with no effect on model accuracy, and a novel two-level privacy approach using local differential privacy together with additive secret sharing, resulting in (ϵ,δ)-differential privacy and protection from gradient inversion and collusion attacks.

Experiments using healthcare monitoring tasks, personal assistants, and image classification reveal that FedSecure:

- Achieves 94.2% accuracy in benchmark datasets with a mere 11.7% drop in case of highly heterogeneous data distribution (α=0.05)

- Provides 40% communication saving over FedAvg due to dynamic parameter freezing
- Guarantees $\varepsilon = 1.5$ differential privacy with minimal accuracy loss, i.e., 1.7%
- Ensures linear scalability for up to 5,000 clients starting from 100 with maintained efficiency in convergence
- Saves resources better by 96.6% in comparison with traditional methods suitable for wearables in IoMT

In terms of comparative analysis, it has been observed that FedSecure surpasses state-of-the-art mechanisms (FedAvg, FedProx, CEFEEL, DDP-SA) regarding privacy, efficiency, and robustness.

Limitations:

- Byzantine attack: Though robust against inference attacks, adversarial resistance is not formally verified yet.
- Diversity of hardware: Training limitations of clients' devices might hamper FL process.
- Round-trip time (RTT): Additional overhead caused by secure aggregation protocol; asynchronous schemes can help tackle this problem.
- Formal regulatory approval: FedSecure needs regulatory compliance validation.

Future Developments:

- Foundational models (FMs): Integration into FL frameworks to accelerate training and personalized FL.
- Blockchain: Tamper-resistant updates for better security and governance.
- Reinforcement learning: Extension into reinforcement learning framework for dynamically changing environments.
- Pilot studies: Deployment on real IoT and wearable devices to test FedSecure on clinical level.
- Post-quantum cryptography: Preparation for future quantum computing challenges.

FedSecure proposes a scalable, efficient, and robust framework for privacy-preserving FL with broad applicability for next-gen AI in areas such as healthcare, mobile services, and intelligent systems, where privacy protection is as important as collaboration.

REFERENCES

[1] E. T. Martínez Beltrán, A. Huertas Celdrán, A. Avilés Serrano, and F. Torres Vega, "CEFEEL: Communication-efficient federated learning for personal assistant applications," IEEE Trans. Consum. Electron., vol. 71, no. 5, pp. 1811–1821, Oct. 2025, DOI: 10.1109/TCE.2025.11216026.

[2] D. Gorbunov, A. D. T. Le, and M. P. L. de Silva, "Foundational models and federated learning: Survey, taxonomy, challenges and practical insights," PeerJ Comput. Sci., vol. 11, no. e2993, Jul. 2025, DOI: 10.7717/peerj-cs.2993.

[3] G. Jain, A. Jain, A. Shukla, and P. Kumar, "AI-driven IoT, federated learning, and surgical robotics: Revolutionizing smart healthcare," in Surgical Robots in Smart Hospitals, A. K. Tyagi, K. Tripathi, S.

Tiwari, and V. Hemamalini, Eds. Hoboken, NJ, USA: Wiley, 2025, ch. 2, DOI: 10.1002/9781394355358.ch2.

[4] W. Wei, F. Nait-Abdesselam, and A. Jammine, "DDP-SA: Scalable privacy-preserving federated learning via distributed differential privacy and secure aggregation," arXiv:2604.07125, Apr. 2026.

[5] E. T. Martínez Beltrán, A. Huertas Celdrán, A. Avilés Serrano, and F. Torres Vega, "NEBULA: A decentralized federated learning platform," PyPI Package, v1.0.0, Jun. 2025. [Online]. Available: https://pypi.org/project/nebula-dfl/

[6] E. Collins and M. Wang, "Federated learning: A survey on privacy-preserving collaborative intelligence," arXiv:2504.17703, Apr. 2025.

[7] S. Chakraborty, R. Gupta, and A. K. Singh, "FedTinyMed: Federated learning enabled tiny multi-task machine learning model for smart healthcare monitoring for IoMT," Comput. Electr. Eng., vol. 123, pp. 109704–109718, Oct. 2025, DOI: 10.1016/j.compeleceng.2025.109704.

[8] R. Kumar, P. Singh, and A. Gupta, "Privacy-preserving and scalable secure aggregation for federated learning in edge computing," in Proc. IEEE Int. Conf. Edge Comput. Smart Syst., Coimbatore, India, Jun. 2025, pp. 1–6, DOI: 10.1109/ECSS.2025.11086126.

[9] M. A. Rahman, F. X. Liu, and C. D. Wright, "Federated learning for big data: A survey on opportunities, applications, and future directions," Expert Syst. Appl., vol. 272, pp. 125801–125822, Dec. 2025, DOI: 10.1016/j.eswa.2025.125801.

[10] N. Jahan, R. Rahman, and M. Wang, "Federated learning: A survey on privacy-preserving collaborative intelligence," arXiv:2504.17703, Apr. 2025.