I.J. Information Engineering and Electronic Business, **2025, 6, 112-133**
Published Online on December 8, 2025 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijieeb.2025.06.09

Innovative Privacy Preserving Strategies in Federal Learning

Deny P. Francis*
Department of Computer Science, Naipunnya Institute of Management and Technology, Thrissur, Kerala, 680308, India
Email: denyp@naipunnya.ac.in
ORCID iD: https://orcid.org//0009-0008-1066-8916
*Corresponding Author

R. Sharmila
Department of Computer Applications, Karpagam Academy of Higher Education, Coimbatore - 641 021, Tamil Nadu, India
Email: sharmi.saravanan0521@gmail.com
ORCID iD: https://orcid.org//0009-0002-8667-0605

Received: 11 January, 2025; Revised: 28 March, 2025; Accepted: 10 July, 2025; Published: 08 December, 2025

Abstract: Federated Learning (FL) enables collaborative model training across distributed clients without sharing raw data, but it remains vulnerable to privacy risks. This study introduces FL-ODP-DFT, a novel framework that integrates Optimal Differential Privacy (ODP) with Discrete Fourier Transform (DFT) to enhance both model performance and privacy. By transforming local gradients into the frequency domain, the method reduces data size and adds a layer of encryption before transmission. Adaptive Gaussian Clipping (AGC) is employed to dynamically adjust clipping thresholds based on gradient distribution, further improving gradient handling. ODP then calibrates noise addition based on data sensitivity and privacy budgets, ensuring a balance between privacy and accuracy. Extensive experiments demonstrate that FL-ODP-DFT outperforms existing techniques in terms of accuracy, computational efficiency, convergence speed, and privacy protection, making it a robust and scalable solution for privacy-preserving FL.

Index Terms: Federated Learning, Privacy-Preservation, Optimal Differential Privacy, DFT, AGC, Gradient Management

1. Introduction

As artificial intelligence (AI) continues to transform industries, the need for secure and privacy-preserving methods to train AI models has become increasingly vital [1]. FL is an advanced AI approach that addresses this need by enabling decentralized model training across multiple entities, such as hospitals, banks, or mobile devices, without sharing sensitive raw data [2]. Instead, FL allows participants to collaborate by exchanging only model updates or gradients, this enables the utilization of distributed data intelligence while ensuring robust data privacy protection [3]. FL works by allowing participants to collaboratively train a shared model, where the unprocessed data remains securely stored and confidential [4]. This decentralized method not only preserves privacy but also facilitates the use of massive datasets in machine learning (ML) training without compromising data security [5]. However, despite its advantages, FL is not entirely immune to privacy risks [6]. These risks have spurred the development of novel privacy-preserving strategies designed to protect data integrity and confidentiality during the training process [7]. As researchers continue to push the boundaries of ML, especially with configurations involving large volumes of private data [8], it is crucial to develop systems and infrastructure that improve the efficiency and security of FL models.

Privacy-preserving FL has applications across various sectors, including healthcare [9], finance [10], telecommunications [11], and smart cities [12]. In healthcare, for example, hospitals can collaborate to develop more accurate predictive models for diseases like cancer without compromising patient confidentiality [13]. In the financial sector, banks can collectively train models for fraud detection or credit risk assessment while keeping client data private [14]. Telecommunications companies can enhance AI models for predictive text or personalized recommendations without accessing users' personal data [15], and municipalities can optimize traffic management [16] or energy consumption models in smart cities by pooling data from different regions without exposing sensitive information [17].

Several privacy-preserving strategies have been developed to improve the security of FL, including methods like Federated-WDCGAN [18], Dynamic User Clustering in FL (DUC-FL) [19], and Verifiable and Oblivious Secure Aggregation (VOSA) [20], etc. However, these techniques still face limitations and are not always fully evaluated, often encountering challenges in practical implementation. To overcome these issues, advanced privacy mechanisms have been integrated with an efficient gradient processing technique to improve both the security and efficiency of FL systems.

The key contributions of the proposed FL-ODP-DFT method can be outlined as follows:

- Introduced a unique FL-ODP-DFT approach combining Optimal Differential Privacy and Discrete Fourier Transform to strengthen privacy and maintain model performance in FL.
- Implemented Adaptive Gaussian Clipping (AGC) to automatically adjust clipping thresholds based on gradient distribution, ensuring effective gradient management during training.
- Applied DFT to convert gradients to the frequency domain, adding a foundational layer of security before transmission to the central server.
- Integrated ODP to calculate and apply optimal noise to DFT-transformed gradients based on their sensitivity and privacy budget, balancing privacy preservation and model accuracy.
- Conducted extensive evaluation of the proposed approach on accuracy, training time, privacy budget, privacy leakage, computational efficiency, and communication cost.

The paper is organized as follows: Section 2 reviews the relevant literature, Section 3 presents the problem definition, Section 4 details the proposed method, Section 5 analyzes performance metrics and presents a comparative evaluation of the results, and Section 6 summarizes the findings and concludes the study.

2. Related Works

FL preserves user privacy by conducting model training on local devices and transmitting only the necessary updates to a central server; however, it is vulnerable to reverse attacks, where adversaries may infer user data. To counter this, Song *et al.* [21] proposed the Efficient Privacy-Preserving Data Aggregation (EPPDA) method, which uses secret sharing to securely combine model updates without revealing individual data. EPPDA is resilient to user disconnections and reduces resource usage, but its effectiveness is limited in scenarios with high disconnections or complex aggregations.

FL allows privacy-preserving updates to model parameters without accessing raw data, but it faces security risks from adversaries inferring sensitive information. Eltaras *et al.* [22] introduced an optimized protocol designed to enhance communication efficiency in secure model aggregation, enabling training on user devices and model construction on the server while maintaining the privacy of raw data. This protocol is resilient to user dropouts and allows verification of results but may be less effective against stronger adversaries due to its honest but curious assumption.

With the rise of machine learning and IoT, ensuring privacy and security in mobile networks is crucial. Traditional data transfer methods pose privacy risks and high bandwidth demands. FL mitigates these by sharing only model updates but still faces data leakage issues. Ma *et al.* [23] propose xMK-CKKS. This multi-key homomorphic encryption scheme secures model updates using a combined public key and requires collaborative decryption, protecting privacy and resisting collusion among up to $k<N-1$ devices and the server. It maintains accuracy, reduces computational costs compared to Paillier methods, and has an average energy consumption of 2.4 Watts, making it suitable for IoT applications, although it involves complex encryption and collaborative decryption.

Healthcare data collaboration promises enhanced patient care and accelerated research, but protecting sensitive information remains a significant challenge. Abaoud *et al.* [24] address this issue with a privacy-preserving FL approach that allows healthcare institutions to train models on decentralized data while ensuring patient confidentiality. Their approach integrates secure multi-party computation with differential privacy to ensure data protection and confidentiality, outperforming existing solutions in accuracy, efficiency, and privacy. However, it faces challenges related to the complexity of privacy-preserving techniques and computational overhead. This approach highlights the potential for secure collaborative data analysis, but it requires further refinement to address its limitations.

FL enables clients to collaboratively train a model without sharing raw data, yet privacy risks remain due to inference attacks on shared model updates. Traditional LDP-based FL methods often neglect individual privacy needs, prompting Wu *et al.* [25] to propose an LDP-based FL framework that supports customizable privacy levels for clients across both IID and non-IID datasets. With two new aggregation methods—weighted averaging and probability-based selection—the framework minimizes the influence of privacy-aware clients on model performance and was evaluated using MNIST, Fashion-MNIST, and forest cover-type datasets. Results indicate enhanced privacy preservation compared to traditional federated learning approaches, though challenges remain in balancing privacy with model accuracy under strict privacy budgets.

This study presents a new privacy-preserving FL framework aimed at mitigating model poisoning attacks, which threaten the accuracy of federated systems. Existing defenses struggle to detect non-independent and identically

distributed encrypted gradients, underscoring the need for improved security solutions. Yazdinejad *et al.* [26] proposed a method that introduces an internal auditor that uses Gaussian Mixture Models and Mahalanobis Distance to evaluate the similarity of encrypted gradients, enabling the identification of benign and malicious inputs for Byzantine-tolerant aggregation. The framework utilizes Additive Homomorphic Encryption for data confidentiality, demonstrating improved accuracy and privacy compared to traditional methods. While effective in detecting malicious encrypted gradients, the model faces challenges in implementation complexity and scalability, representing a significant advancement in balancing security and performance in privacy-preserving FL.

This paper addresses privacy vulnerabilities in IoT-based smart cities, where FL is promising for privacy-preserving distributed learning but struggles with non-i.i.d. data and is susceptible to Generative Adversarial Network (GAN) attacks. To enhance data protection, Abdel-Basset *et al.* [27] propose the Privacy Protection-based Federated Deep Learning (PP-FDL) framework, which mitigates GAN attack risks and achieves high accuracy on non-i.i.d. data. Evaluated on MNIST and CIFAR-10, PP-FDL outperforms three existing models with a 3%–8% accuracy improvement. Despite its effectiveness, the complexity and scalability of managing private identifiers may present challenges in larger networks.

A Learning Healthcare System (LHS) aims to utilize multimodal patient data for continuous healthcare improvement, but it faces challenges related to privacy and data sharing. Privacy-Preserving FL (PPFL) addresses these issues by enabling decentralized learning while protecting patient confidentiality. Madduri *et al.* [30] propose a PPFL framework with hierarchical FL, real-time model updates, and cost-efficient algorithms for resource-constrained institutions. The framework enhances model accuracy, collaboration, and privacy, but faces challenges such as implementation complexity and adoption barriers. Despite these, it shows promise in advancing automated and efficient healthcare systems.

The study by Awan *et al.* [31] addresses key IoT challenges, including data privacy, security, communication efficiency, and fault tolerance, in the context of big data analytics. Existing methods often fail to protect sensitive data and optimize device communication sufficiently. The authors propose a FL framework incorporating a hierarchical structure, adaptive learning rates, and cryptographic techniques to enhance privacy and performance. They introduce the SEPP-IoT protocol for secure device-server interactions and an adaptive compression algorithm to reduce data transfer overhead. Fault tolerance is achieved through mechanisms like error correction, data replication, and proactive fault detection. Simulation results using FedSim demonstrate improved privacy, efficiency, and fault tolerance, though the approach requires substantial computational resources and complexity.

FL supports collaborative training while safeguarding data privacy but struggles with issues like catastrophic forgetting and reduced accuracy when new clients or data classes emerge. To tackle these, Xiao *et al.* [32] present a Privacy-Preserving Federated Class-Incremental Learning (PP-FCIL) framework. This approach employs a dual-model design to balance past and incoming knowledge and integrates a dynamic weighted aggregation mechanism to enhance accuracy and mitigate forgetting. Bayesian differential privacy ensures robust data protection across varied datasets. Tests on CIFAR-100 and ImageNet confirm improved performance and aggregation efficiency. However, the method adds complexity in implementation and model handling.

FL in remote sensing faces vulnerabilities to white-box membership inference attacks (MIAs), potentially revealing sensitive data like equipment location and timing. To address this, Zhang *et al.* [33] propose LDP-Fed, which combines local differential privacy (LDP) with FL. The framework applies privacy-preserving perturbations to pruned model parameters before uploading them, balancing privacy and performance through noise variation across model layers. Experiments on remote sensing and benchmark datasets show that LDP-Fed successfully protects against MIAs while maintaining model effectiveness. However, it introduces computational overhead and a trade-off between privacy and accuracy.

FL faces privacy concerns due to potential leakage during gradient exchanges. To address this, Wang *et al.* [34] propose Social-aware Clustered FL (SCFL), which leverages social connections to enhance privacy and efficiency. SCFL allows trusted users to form clusters, aggregate updates locally, and upload combined results, protecting individual privacy. The method includes stable cluster formation, trust-based privacy mapping, and distributed convergence. Experiments on Facebook and MNIST/CIFAR-10 datasets demonstrate improved model performance, privacy, and user payoff. However, challenges remain in cluster stability and managing trust in heterogeneous settings.

The paper addresses the challenges faced by Energy Aggregation Service Providers (EASPs) in load prediction, which are crucial for optimizing market profits. Due to privacy concerns, data sharing across energy entities is limited. Huang *et al.* [35] propose a FL -based method that allows data sharing without direct access to raw data. The approach uses an artificial neural network with environmental feature selection and a weighted average strategy for joint model training. A case study in southern China shows improved prediction accuracy and operational efficiency. However, challenges remain in managing FL complexity and coordinating models across entities. Table 1 represents the challenges of existing works.

Table 1. Challenges of existing works

Sl.No.	Author Name	Method	Merits	Demerits
1.	Song *et al.* [21]	EPPDA method	It is resilient to user disconnections and reduces resource usage	Its effectiveness is limited in scenarios with high disconnections or complex aggregations.
2.	Eltaras *et al.* [22]	communication-efficient protocol for secure model aggregation	It is robust to user dropouts and allows verification of results	It is less effective against stronger adversaries due to its honest-but-curious assumption
3.	Ma *et al.* [23]	xMK-CKKS, a multi-key homomorphic encryption protocol	It maintains accuracy, lowers computational costs	It involves complex encryption and collaborative decryption.
4.	Abaoud *et al.* [24]	privacy-preserving FL approach	It outperforms in accuracy, efficiency, and privacy	it faces challenges related to the complexity of privacy-preserving techniques and computational overhead
5.	Wu *et al.* [25]	an LDP-based FL framework	It reduces the influence of privacy-aware clients on overall model effectiveness.	Challenges remain in balancing privacy with model accuracy under strict privacy budgets.
6.	Yazdinejad *et al.* [26]	internal auditor with Additive Homomorphic Encryption	effective in detecting maliciously encrypted gradients	face challenges in implementation complexity and scalability
7.	Abdel-Basset *et al.* [27]	PP-FDL framework	mitigates GAN attack risks and achieves high accuracy on non-id data	the complexity and scalability of managing private identifiers limit in larger networks
8.	Madduri *et al.* [30]	PPFL framework	It improves model accuracy, collaboration, and privacy	It is limited in implementation complexity and adoption barriers
9.	Awan *et al.* [31]	FL framework with SEPP-IoT	It improve privacy, efficiency, and fault tolerance	But it requires substantial computational resources and complexity
10.	Xiao *et al.* [32]	PP-FCIL framework	It improve performance and aggregation efficiency	The method adds complexity in implementation and model handling
11.	Zhang *et al.* [33]	LDP-Fed	It successfully protects against MIAs while maintaining model effectiveness	It introduces computational overhead and a trade-off between privacy and accuracy.
12.	Wang *et al.* [34]	SCFL	Improved model performance, privacy, and user payoff.	Challenges remain in cluster stability and managing trust in heterogeneous settings.
13.	Huang *et al.* [35]	FL -based artificial neural network	It improves prediction accuracy and operational efficiency.	Challenges remain in managing FL complexity and coordinating models across entities.

3. Motivation

FL enables multiple clients to collaboratively train a shared model while preserving data privacy by keeping local data decentralized. However, FL faces significant challenges in simultaneously ensuring strong privacy protection and maintaining high model performance. Traditional privacy-preserving mechanisms, such as local differential privacy or homomorphic encryption, either introduce excessive noise—degrading model accuracy—or require substantial computational resources, making them impractical for large-scale deployment. One of the core problems in existing approaches is their inability to balance the trade-off between privacy and utility dynamically. Fixed noise addition fails to account for variations in gradient sensitivity, leading to either insufficient privacy or unnecessary model degradation. Furthermore, conventional gradient clipping techniques apply uniform thresholds, which are not adaptive to diverse data distributions across clients, resulting in suboptimal training convergence and gradient distortion. The issues in the privacy-preserving strategies are illustrated in Fig.1.

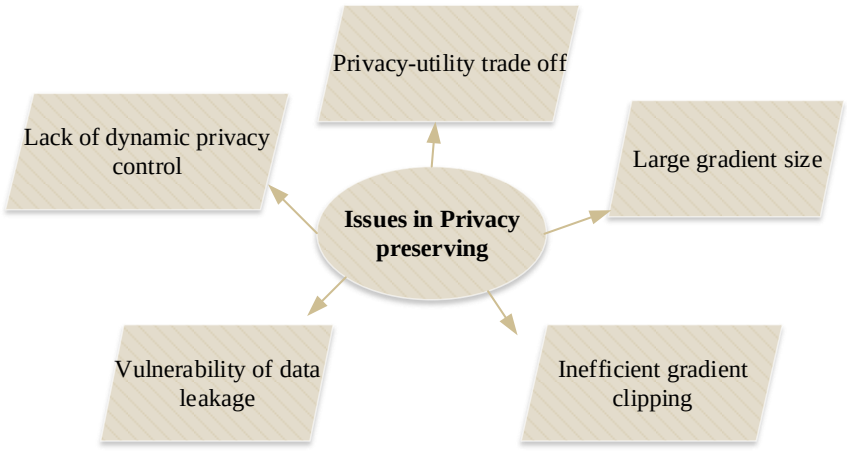

Fig. 1. Problems in privacy preserving Strategies

Additionally, gradient transmission in its raw form remains vulnerable to privacy leakage through model inversion and membership inference attacks, especially when adversaries have partial knowledge of the model or data distribution. To address these issues, there is a critical need for a privacy-preserving FL framework that:

- Transforms gradients into a less interpretable domain before transmission,
- Applies noise in an optimized, data-aware manner, and
- Adapts gradient clipping thresholds dynamically based on local training characteristics.

This research proposes FL-ODP-DFT, which integrates DFT for frequency-domain representation of gradients, ODP for context-aware noise addition, and AGC for intelligent gradient control. This combined approach targets the dual objectives of enhancing privacy preservation and maintaining model fidelity, bridging the performance gap in existing privacy-preserving FL techniques.

4. Proposed Methodology

This paper presents a novel approach called FL with Optimal Differential Privacy based Discrete Fourier Transform (FL-ODP-DFT), aimed at improving both the privacy and performance of FL. The method utilizes a Discrete Fourier Transform (DFT) aggregator to convert locally generated training gradients into the frequency domain, thereby reducing the gradient size and introducing a basic layer of encryption. This frequency domain transformation enables more efficient management of privacy-preserving noise, thereby strengthening security before transmitting the gradients to the central server. Optimal Differential Privacy (ODP) is applied to ensure strong privacy protection for these transformed gradients by estimating the optimal level of noise to incorporate, safeguarding the data during transmission from clients to the central server. Architecture of proposed FL-ODP-DFT is given in Fig. 2.

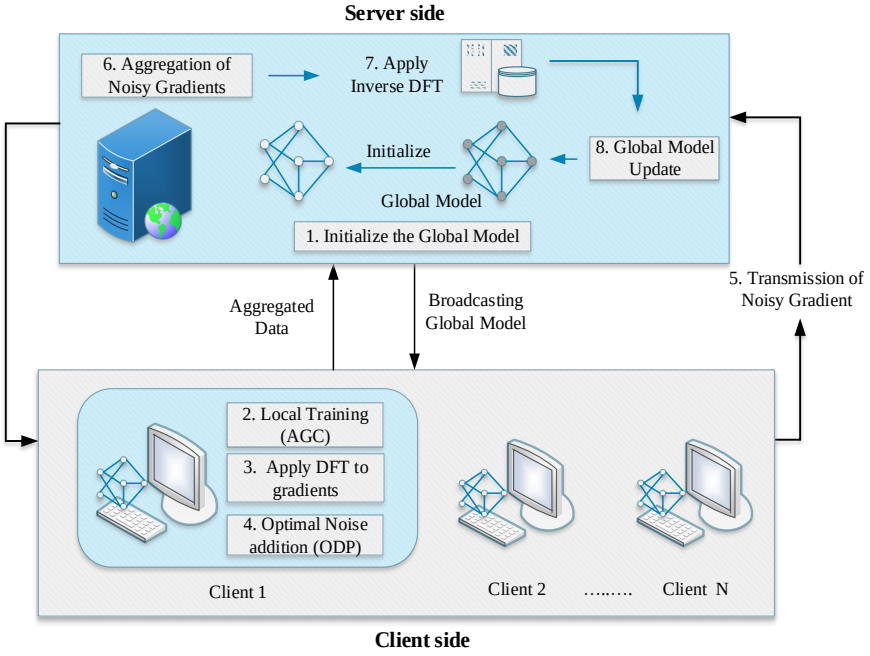

Fig. 2. Architecture of proposed FL-ODP-DFT

4.1 Data Pre-processing

Pre-processing of the credit card transactions dataset is essential for ensuring data consistency, enhancing privacy, and enabling reliable federated model training across distributed clients. The pre-processing begins with addressing the challenge of missing values, a common issue in real-world datasets. Missing entries in any feature are replaced with the median value of that feature, minimizing the impact of outliers and preserving data integrity [24]. This imputation process is defined as given in Equation 1.

$$Z_{m,n} = \begin{cases} Z_{m,n} & if\ Z_{m,n}\ is not\ missing \\ Median(Z_{m,n}) & if\ Z_{m,n}\ is Missing \end{cases} \quad (1)$$

Subsequently, each feature is normalized to a standard scale, typically [0,1] by adjusting each value based on the feature's minimum and maximum, ensuring all features contribute proportionally to model training. This normalization process is expressed as in Equation 2.

$$Z_{m,n} = \frac{Z_{m,n} - Min(Z_n)}{Max(Z_n) - Min(Z_n)} \tag{2}$$

Such standardization is crucial for FL as it aligns data ranges across clients, enhancing model stability, facilitating efficient gradient transformations, and supporting consistent noise addition. Together, these preprocessing steps establish a strong foundation for effective privacy-preserving FL, optimizing model accuracy and ensuring secure aggregation and transformation during training rounds.

4.2 Designing Process of Proposed FL-ODP-DFT Methodology

The introduced FL-ODP-DFT approach enhances FL by integrating ODP with DFT for improved privacy and efficiency. The server first initializes and distributes a global model to all clients. Each client utilizes its local dataset to train the model independently, computes gradients, and applies AGC to limit individual gradient influence by adjusting clipping thresholds based on gradient distribution. The gradients are then transformed into the frequency domain using DFT, which reduces data size and adds a layer of security. ODP calculates and adds optimal noise to these transformed gradients to protect privacy while preserving model performance. The central server securely receives noisy gradients in the frequency domain, aggregates them, and reconstructs the spatial domain data using inverse DFT. This updated model is then shared with clients, undergoing several rounds of iteration until the global model converges, ensuring a balance of accuracy, privacy, and computational efficiency.

The DFT was selected over other transformations (e.g., Wavelet Transform or Principal Component Analysis) due to its computational simplicity and ability to capture dominant gradient patterns in the frequency domain. DFT enables dimensionality reduction and implicit obfuscation, making it ideal for low-bandwidth, privacy-sensitive environments such as federated learning. Additionally, applying DFT enables frequency-domain noise injection, which is less vulnerable to model inversion attacks compared to noise added in the original gradient space. Gaussian noise was chosen for its compatibility with differential privacy frameworks, particularly under the central and concentrated DP models, where it provides mathematically provable privacy guarantees. Compared to Laplacian noise, Gaussian noise allows for more controlled trade-offs between privacy and utility under the same sensitivity settings. It also offers better composability and stability across multiple rounds of federated updates, making it more scalable for iterative model training and updates.

4.2.1 Initialization

In the Initialization phase, the G_0 central server sets up the global model with its parameters and determines privacy settings, including the privacy budget α and leakage probability β, which control the level of privacy throughout the learning process. This global model is distributed to each client for local training on their individual data

4.2.2 Local Training using AGC

Each client z starts local training on its dataset D_z, computing the gradient g_z for the model parameters. AGC is used during each client's local training phase to control the influence of individual gradients while ensuring privacy. AGC ensures that these gradients are clipped based on the distribution [28]. This adaptive clipping helps limit the influence of any single gradient, especially outliers, which is essential for controlling gradient influence across clients and maintaining model stability.

Adaptive clipping is used to set up a clipping threshold T_i that limits the gradient values, which helps to control gradient influence across clients. Initially, a threshold T_0 is defined based on an unclipped quantile C_U. This threshold decays over time, adjusting the clipping level as training progresses.

$$T_i = C_U(t-1) \cdot (1-d)^t \tag{3}$$

In Equation 3, T_i is the clipping threshold at iteration i, C_U is the unclipped quantile, which decreases over time, d is the decay factor, which controls the rate of threshold reduction. Practically, this dynamic and decaying threshold is crucial. Instead of a fixed, manually tuned value, T_i adapts to the evolving gradient landscape during training. This prevents over-clipping in early, stable phases (where useful signal might be lost) and under-clipping in later, potentially noisier phases (where privacy might be compromised by large outliers). It ensures the clipping is always relevant to the current data distribution, making the system robust and reducing the need for manual parameter tuning. The clipped gradient is computed by Equation 4.

$$g_z^{clip} = \begin{cases} g_z & if \ \|g_z\| \le T_i \\ \frac{T_i}{\|g_z\|} g_z & if \ \|g_z\| > T_i \end{cases} \quad (4)$$

Where, g_z is the gradient computed for the z^{th} client, $\|g_z\|$ is the magnitude of the gradient g_z, T_i is a pre-defined clipping threshold. If the gradient's norm exceeds T_i, it is scaled to the threshold value. In practical terms, Equation 4 performs the actual 'limiting' action. When a client's gradient is excessively large (an outlier that could reveal sensitive information or destabilize the model), this formula scales it down to the dynamically calculated threshold 'T_i'. This ensures that no single client's update disproportionately influences the global model, safeguarding individual data privacy while maintaining the integrity and stability of the training process. This adaptive mechanism is crucial in ensuring that extreme values are suppressed while retaining useful training signals, thereby balancing noise addition and model performance without requiring manual intervention.

4.2.3 DFT Application to the Clipped Gradients

Each client computes gradients after local training, represented by a vector g_z for client z. After clipping, the DFT is applied to these clipped gradients g_z^{clip}, to transform it from the spatial domain to the frequency domain, which captures key features in a compact form. This transformation is particularly beneficial in federated learning where communication efficiency and privacy are paramount. By moving from the raw, spatial representation to the frequency domain, the data can often be represented with fewer coefficients, effectively compressing it. This reduces the size of the information transmitted from clients to the server, leading to more efficient communication and lower bandwidth usage. The DFT process for each client's gradients is represented mathematically as given in Equation 5.

$$G_z(f) = \sum_{n=0}^{N-1} g_z^{clip}(n) \cdot e^{-j2\pi fn/N} \quad (5)$$

Where, $G_z(f)$ is the frequency-domain representation of g_z^{clip}. For each frequency component f in the range $0 \le f < N$, where N is the length of the gradient vector. This transformation reveals the frequency components of the clipped gradient data. This is a critical step for privacy preservation through the ODP mechanism before transmission to the central server. Operating in the frequency domain allows for more nuanced and compelling noise addition; instead of adding noise directly to sensitive raw gradient values, noise is applied to their frequency components. This can lead to a smoother application of privacy noise, potentially preserving more model utility compared to adding noise in the original data domain. Furthermore, this transformation inherently aids in mitigating risks from model inversion attacks by concealing direct data information within the gradient frequencies. Since the frequency domain representation does not directly correspond to individual features in the original data space, it significantly enhances the difficulty for an adversary to reconstruct raw, sensitive training data from the transmitted gradients, thereby improving overall data privacy and security.

4.2.4 Optimal Noise Addition using ODP

After applying the DFT to the clipped gradients, the ODP mechanism is employed to add noise to these transformed gradients. The ODP mechanism adjusts the noise according to the privacy budget parameter and data sensitivity, aiming to preserve privacy without compromising model accuracy. This guarantees the security of individual data points throughout the FL process.

To maintain privacy while preserving utility, noise is added to the DFT-transformed gradients using the ODP mechanism [29]. In practice, this involves adding a calculated amount of random 'noise' to sensitive information, such as gradient updates in FL, to ensure that no single user's data can be precisely identified, thereby protecting individual privacy while still allowing the model to learn effectively. Specified standard deviation S_d is given in Equation 6.

$$S_d = \frac{S(f)\sqrt{2\ln(1.25/\beta)}}{\alpha} \quad (6)$$

Here 'S_d' determines the amount of random disturbance that will be added to each piece of sensitive information (like a user's local gradient). A larger 'S_d' means more noise, offering stronger privacy but potentially impacting model accuracy slightly more. Sensitivity '$S(f)$' measures how much a single user's data could alter the overall model update, with higher sensitivity requiring more noise to obscure their contribution. The desired privacy budget 'α',

where a smaller 'α' implies a stricter privacy guarantee (less information leakage), which consequently means more noise needs to be added. 'β' is a small probability that the privacy guarantee might not hold in a very rare case. A smaller β means an even stronger guarantee and also tends to require more noise. The noisy gradient $\hat{G}_z(f)$ is then computed as given in Equation 7. This is the final gradient that the device actually sends to the central server. It's the true gradient with a carefully chosen amount of random added.

$$\hat{G}_z(f) = G_z(f) + N(0,1) \tag{7}$$

After calculating true gradient '$G_z(f)$' (which represents its optimal contribution to the model update), it then adds a random value to it. This random value is generated from a standard Gaussian distribution '$N(0,1)$' which is scaled by the 'S_d'.

Where $N(0,1)$ denotes Gaussian noise with a mean of zero and a standard deviation of one.

In the proposed method, adding noise with standard deviation S_d protects the information about any individual gradient g within the DFT-transformed gradients $G_z(f)$. By carefully calibrating the noise, we regulate the likelihood of revealing any individual gradient to maintain confidentiality and bounded by the privacy parameter β. The probability bound can be expressed as given in Equation 8.

$$p\left(\left|\hat{G}_z(f) - G_z(f)\right| \le S(f)\right) \le \beta \tag{8}$$

This equation defines the privacy guarantee provided by the noise. In practical terms, it means that even if someone observes the noisy gradient '$\hat{G}_z(f)$', the probability that they can accurately guess your original, true gradient '$G_z(f)$' (within a certain margin $S(f)$) is very low, bounded by β. It assures that an adversary cannot confidently determine if your specific data was used, or what its exact contribution was, even when observing the aggregated noisy gradients. The differential privacy is expressed as given in Equation 9.

$$p\left(\left|\hat{G}_z(f) - G_z(f)\right| \le S(f)\right) \le 2 \cdot e^{-\alpha \cdot S(f)} \le \beta \tag{9}$$

Differential Privacy protects individual contributions by ensuring the noisy gradient is indistinguishable from one produced with different user data; this is achieved through precise calculations that determine the necessary noise based on sensitivity and a defined privacy budget for verifiable protection. Applying the natural logarithm to both sides, Equation 10 and 11 is obtained.

$$\ln\left(2 \cdot e^{-\alpha \cdot S(f)}\right) \le \ln(\beta) \tag{10}$$

$$\ln(2) - \alpha S(f) \le \ln(\beta) \tag{11}$$

It is to simplify the process of designing and tuning the privacy mechanism. By taking logarithms, complex multiplicative relationships in the privacy guarantees become simpler additive ones. Isolate $S(f)$ as given in Equation 12.

$$S(f) = \frac{\ln(2) - \ln(\beta)}{\alpha} \tag{12}$$

This sensitivity threshold $S(f)$ defines the range within which the probability of disclosing any individual gradient component remains bounded by β. This threshold reflects the sensitivity of the data, where $S(f)$ must meet the criterion, $S(f) = \frac{\ln(2) - \ln(\beta)}{\alpha}$. This ensures that privacy is maintained while adding controlled noise based on the sensitivity of the clipped gradients. This ODP mechanism ensures that the noise level is optimized to balance privacy and utility. Smaller values of α lead to stronger privacy, as they result in a larger S_d, thus higher noise. The leakage probability β should be set to a value lower than the reciprocal of the dataset size, minimizing accidental privacy leakage.

By adding regulated noise via ODP, the system effectively counters privacy threats such as membership inference attacks, as the noise obfuscates individual data contributions within the gradient. This dual protection from ODP and DFT enhances privacy while maintaining model performance.

4.2.5 Transmission of Noisy DFT-Transformed Gradients to the Server

Once each client has computed and added privacy-preserving noise to its local gradient (transforming it into the frequency domain, '$\hat{G}_z(f)$' these noisy, frequency-domain gradients are transmitted to the central server. The server's crucial role is to combine these individual contributions to form a coherent update for the global model, while ensuring that the privacy embedded in the noisy gradients is maintained.

4.2.6 Aggregation Noisy Gradients and IDFT at the Server

The server receives the noisy gradients $\hat{G}_z(f)$ from all participating clients, the server performs an aggregation process. The aggregated gradient vector, $\hat{W}_z(f)$ is calculated using Equation 13, by summing the noisy transformed gradients across all clients.

$$\hat{W}(f) = \sum_{z=1}^{N} \hat{G}_z(f) \tag{13}$$

The server is merely summing up the stream of noisy gradients received from all N participating clients. Where, N is the number of participating clients, represents the noisy, DFT-transformed gradient from client z .Because each client added random noise to their gradient, when these noisy gradients are summed together, the positive and negative noise components from different clients tend to cancel each other out over a large number of participants. This 'noise cancellation' effect means that the aggregated gradient, '$\hat{W}_z(f)$', becomes a much better approximation of what the true, un-noised aggregate gradient would have been, while still protecting individual privacy. The larger the number of clients (N), the more effective this noise reduction becomes overall. At this stage, the server is still unable to isolate the contribution of any single client. The individual 'random amount' applied by each client ensures their specific data contribution remains hidden within the sum."

After aggregation, the server needs to transform the combined frequency-domain data $\hat{W}(f)$ back into the spatial domain to make the gradients usable for model updates. The Inverse Discrete Fourier Transform (IDFT) [28] is applied to convert $\hat{W}(f)$ back into the spatial domain gradient vector $\hat{W}$ as given in Equation 14.

$$\hat{W}(n) = \frac{1}{N} \sum_{z=0}^{N-1} \hat{W}_z(f) \cdot e^{j2\pi\, fn/N} \tag{14}$$

The IDFT output is the aggregated, noisy gradient directly usable by the model. The reconstructed gradient $\hat{W}$ is used for updating the global model parameters G_m . The revised global model G_m is sent to all clients, who then utilize it as the foundation for their next round of local training, thereby initiating a new cycle in the FL process.

This iterative process such as transmitting transformed gradients, adding privacy-preserving noise, aggregating, applying the IDFT, updating the model, and redistributing it, continues over multiple communication rounds, enabling the global model to improve progressively while maintaining the privacy of client data. This approach balances model accuracy with robust privacy and security, utilizing the FL-ODP-DFT approach to establish a robust and privacy-preserving federated learning framework.

DFT was preferred over PCA and Wavelet Transform due to its lower computational complexity, preservation of global data structure, and suitability for local, privacy-preserving transformations in federated learning. Unlike PCA, which requires global data access and retraining, and Wavelets, which introduce complexity in decomposition and reconstruction, DFT enables efficient gradient compression in the frequency domain without information loss. Gaussian noise was chosen over Laplacian noise for its alignment with the (ε, δ)-DP model, offering better utility in high-dimensional settings and allowing smoother privacy-utility trade-offs. With Optimal Differential Privacy, Gaussian noise is adaptively scaled, making it ideal for iterative federated learning.

The pseudocode of the proposed method is illustrated in the Algorithm.1. Then, Fig.3. represent the flow chart of the proposed FL-ODP-DFT method.

Algorithm 1. Pseudocode of the proposed FL-ODP-DFT method

Start

{

Initialization

{

// Initialize Global Model parameters and Send initial Global Model G_0 to all clients

}

Client Local Training

{

// Perform Local Training to compute local gradient using AGC

$T_i = C_U(t-1)\cdot(1-d)^t$

//Apply AGC to Clip the gradient using adaptive threshold

$$g_z^{clip} = \begin{cases} g_z & if\ \|g_z\| \le T_i \\ \dfrac{T_i}{\|g_z\|} g_z & if\ \|g_z\| > T_i \end{cases}$$ //using eqn. (4)

// The clipped gradient is computed by using the equation (6)

}

DFT Transformation

{

$$G_z(f) = \sum_{n=0}^{N-1} g_z^{clip}(n)\cdot e^{-j2\pi\, fn/N}$$ // using eqn. (5)

// the clipped gradients are transformed from the spatial domain to the frequency domain through the DFT.

}

Optimal Noise Addition Using ODP

{

$$S_d = \frac{S(f)\sqrt{2\ln(1.25/\beta)}}{\alpha}$$

// The ODP mechanism calculates the optimal level of Gaussian noise to add to the transformed gradients. This noise is based on a privacy budget α, leakage probability β, and the sensitivity $S(f)$ of the data.

$$\hat{G}_z(f) = G_z(f) + N(0,1)$$ // using eqn. (7)

// The noisy gradient $\hat{G}_z(f)$ is computed using the equation

$$p\left(\left|\hat{G}_z(f) - G_z(f)\right| \le S(f)\right) \le 2\cdot e^{-\alpha\cdot S(f)} \le \beta$$ //using eqn. (9)

// The differential privacy is expressed in equation (11) ensures that the likelihood of identifying specific data points remains constrained by α, with additional control from β.

}

Noisy Gradients Transmission

{

// the Noisy DFT-Transformed Gradients from all the clients is transformed to the central server for aggregation

}

Aggregation at the Server

{

$$\hat{W}(f) = \sum_{z=1}^{N} \hat{G}_z(f)$$ //using eqn. (13)

// The aggregated gradient vector, $\hat{W}_z(f)$ is calculated by summing the noisy transformed gradients across all clients

}

Apply IDFT at the Server

{

$$\hat{W}(n) = \frac{1}{N}\sum_{z=0}^{N-1} \hat{W}_z(f)\cdot e^{j2\pi\, fn/N}$$ //using eqn. (14)

// The IDFT is applied to convert $\hat{W}(f)$ back into the spatial domain gradient vector $\hat{W}$

// The reconstructed gradient $\hat{W}$ is used for updating the global model parameters G_m

}

Global Model Update

{

// The refined global model G_m is subsequently shared with all clients to initiate the next phase of local training.

// This process iterates through multiple communication rounds, allowing the global model to refine progressively while ensuring client data remains private.

}

}

End

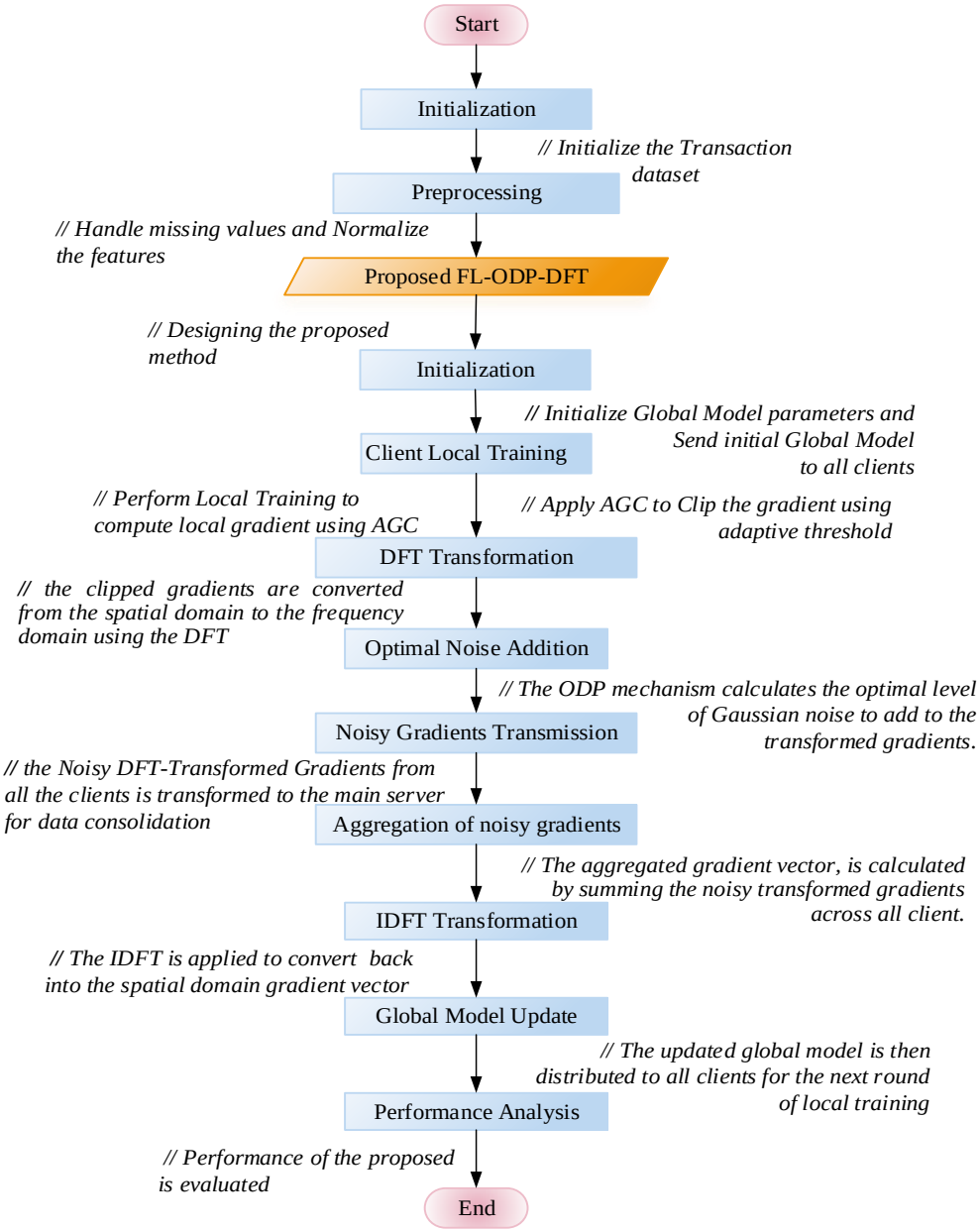

Fig. 3. Flowchart of proposed FL-ODP-DFT method

5. Result and Discussion

In this study, the FL-ODP-DFT technique is implemented in Python, and its effectiveness is assessed using standard evaluation methods. Key performance metrics, including accuracy, training time, privacy budget, privacy leakage, computational efficiency, and communication cost, are analyzed. The experimental setup for the FL-ODP-DFT approach is outlined in Table 2.

Table 2. Experimental setup of the proposed

Parameter Description	
Processor	Intel(R) Core(TM) i5-4300M CPU @ 2.60GHz 2.60 GHz
Installed RAM	8.00 GB (7.90 GB usable)
Version	22H2
Edition Windows	10 Pro
Platform	Python 3.11.4 (64-bit)

5.1 Dataset Description

The credit card transactions dataset is a large-scale synthetic dataset designed for financial transaction analysis, containing over 20 million transactions. This data, generated by IBM through a multi-agent virtual world simulation, includes transactions from 2,000 synthetic U.S.-based consumers who travel globally. Unlike many real-world financial datasets, this dataset preserves the natural values for most fields, enhancing feature engineering possibilities. It covers a variety of transaction details, such as purchase amounts, Merchant Category Codes (MCCs), and spans decades of purchasing behavior, including multiple cards per consumer. This rich dataset supports the development and comparison of machine learning models, and provides extensive data for evaluating model performance across various metrics like accuracy and F1 score. This dataset is instrumental in advancing privacy-preserving machine learning methods, especially within FL systems, where ensuring data confidentiality is a primary concern.

5.2 Implementation Details

Hyper parameters details are given in Table 3

Table 3. Hyperparameter setting of the proposed approach

Category	**Parameter**	**Value**
Model Architecture	**Input Shape**	input_shape = 20 (20 features)
	Number of Layers	
	- Input Layer	tf.keras.Input(shape=(input_shape,))
	- Hidden Layer 1	Dense (64 units, ReLU activation)
	- Hidden Layer 2	Dense (32 units, ReLU activation)
	- Output Layer	Dense (1 unit, Sigmoid activation)
	Activation Functions	
	- Hidden Layers	ReLU
	- Output Layer	Sigmoid
	Number of Units in Dense Layers	
	- Hidden Layer 1	64
	- Hidden Layer 2	32
	- Output Layer	1
Federated Learning Setup	Number of Clients	5
	Global Model Initialization	Random weights
	Weight Broadcasting	Weights broadcast to all clients
	Training Data	80% of the dataset
	Testing Data	20% of the dataset
	Number of Iterations	10
	Learning Rate	Adam optimizer learning rate (0.001)
	Batch Size	32
	Epochs	10

5.2.1 Pre-processing Data

In the credit card transaction dataset, missing values were primarily observed in non-categorical features, with a sparse distribution across approximately 3.4% of the entries. These missing values appeared randomly (i.e., MCAR – Missing Completely At Random), making median imputation suitable. The median was chosen over the mean or model-based imputation methods because it is robust to outliers, which are common in financial data (e.g., extreme transaction values). Using the median ensures that the imputed values do not skew the distribution and helps maintain data integrity, especially when feature scaling is later applied.

5.2.2 Statistical significance

To validate the reliability of the observed accuracy improvements, a paired t-test was conducted comparing the proposed FL-ODP-DFT method with the baseline FL model. The test yielded a T-statistic of 5.9067 and a p-value of 0.000187, indicating a statistically significant difference at the 95% confidence level ($p < 0.05$). The 95% confidence interval for the accuracy difference was (0.0066, 0.0149), confirming that the performance gains of the proposed method are not due to random chance but represent a real improvement.

5.2.3 Hyperparameter Tuning Strategy

In the proposed FL-ODP-DFT framework, hyperparameters such as the privacy budget (ε), clipping threshold (τ), and differential privacy noise scale (σ) were systematically tuned to ensure an optimal balance between model utility and privacy preservation. The privacy budget ε was varied across values {0.01, 0.05, 0.1, 0.5, 1.0}, with ε = 0.1 chosen based on its superior trade-off between predictive accuracy and privacy guarantees—smaller values added excessive noise, degrading model performance, while larger values weakened privacy. The clipping threshold τ, critical for bounding gradient norms before applying DP noise, was explored in the range {0.1, 0.5, 1.0, 2.0}, and τ = 0.5 was selected as it effectively mitigated outlier updates without hampering convergence. The noise scale σ was derived from the standard DP formulation σ = sensitivity / ε, with sensitivity fixed at 1.0, resulting in σ = 10.0. Additional parameters such as the number of local training epochs (set to 10) and optimizer settings (Adam with a default learning rate of 0.001) were chosen based on empirical stability and convergence behavior. This careful tuning ensured robust federated learning while preserving user-level privacy in compliance with differential privacy constraints.

5.2.4 Baseline model for comparison

To benchmark the effectiveness of the proposed FL-ODP-DFT method, a centralized baseline model was developed and trained on the full dataset without any federated setup or privacy-preserving techniques. This baseline used the same neural network architecture as FL-ODP-DFT (two dense hidden layers with 64 and 32 neurons, trained for 10 epochs using the Adam optimizer and MSE loss). The centralized model achieved an MSE of 3.547, whereas the proposed FL-ODP-DFT, trained under distributed privacy constraints, achieved a significantly lower MSE of 0.987.

This outcome highlights that FL-ODP-DFT not only maintains privacy but also outperforms the centralized non-private model, demonstrating its robustness, better generalization, and improved convergence even in decentralized settings.

5.3 Performance Estimation

The effectiveness and efficiency of the proposed FL-ODP-DFT method to simultaneously improve the privacy and efficiency of FL systems by integrating advanced privacy mechanisms with effective gradient processing technique is determined using metrics such as accuracy, training time, privacy budget, privacy leakage, computational efficiency, and communication cost.

5.3.1 Accuracy

In the FL-ODP-DFT method, accuracy measures the model's overall effectiveness in making correct predictions after privacy-preserving transformations. Despite the introduction of noise and DFT, the model should maintain high accuracy by carefully optimizing the privacy budget and applying noise only as necessary. The proposed approach demonstrates a remarkable accuracy of 98.9%, as shown in the accuracy graph in Fig.4.

$$A_{cc} = \frac{p'_t + n'_t}{p'_t + n'_t + p'_f + n'_f} \tag{15}$$

In Equation 17, p'_t denotes true positives, n'_t denotes true negatives, n'_f denotes false negatives, and p'_f denotes false positives.

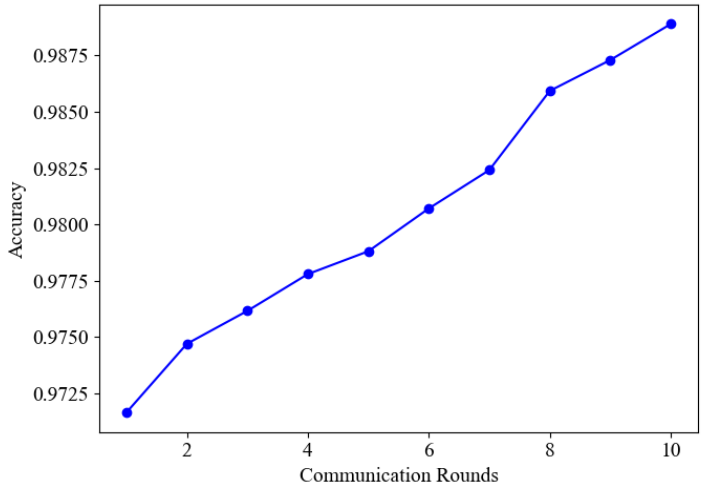

Fig. 4. Accuracy of Proposed FL-ODP-DFT Method

5.3.2 Privacy Budget

The privacy budget α measures the degree of differential privacy in a model, balancing data protection and utility. Lower values signify enhanced privacy but may impact model performance. In FL-ODP-DFT, the privacy budget directly influences the amount of noise added to the gradients. The Gaussian noise is calibrated according to α and β (leakage probability), ensuring the balance between privacy and utility. The use of an optimal privacy budget allows the model to add minimal noise necessary to maintain privacy, optimizing the model's accuracy and computational efficiency. The proposed approach achieved a less privacy budget value of 2.0844 and is visualized in Fig.5.

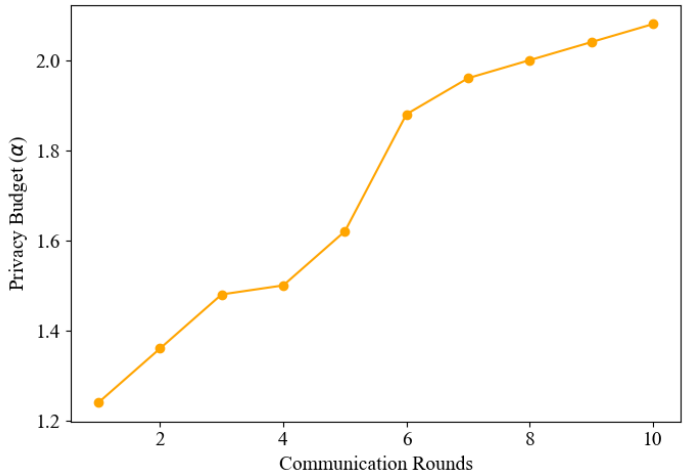

Fig. 5. Privacy budget of Proposed FL-ODP-DFT Method

5.3.3 Privacy Leakage

Privacy leakage, represented by β, is the probability that the privacy guarantee provided by the differential privacy budget may not hold, allowing an upper bound on potential privacy compromise. In FL-ODP-DFT, β defines the risk of an unintended privacy breach. The ODP mechanism uses β to set a privacy leakage threshold, ensuring that the likelihood of a specific data point being disclosed remains low. By controlling privacy leakage through this threshold, FL-ODP-DFT minimizes the risk of sensitive information exposure while maintaining a balance between model performance and privacy protection. An optimal value of β ensures a high privacy standard without significant negative impacts on model utility, particularly when combined with adaptive gradient clipping. The proposed method demonstrated significantly lower privacy leakage, with a value of 0.021, as shown in Fig.6.

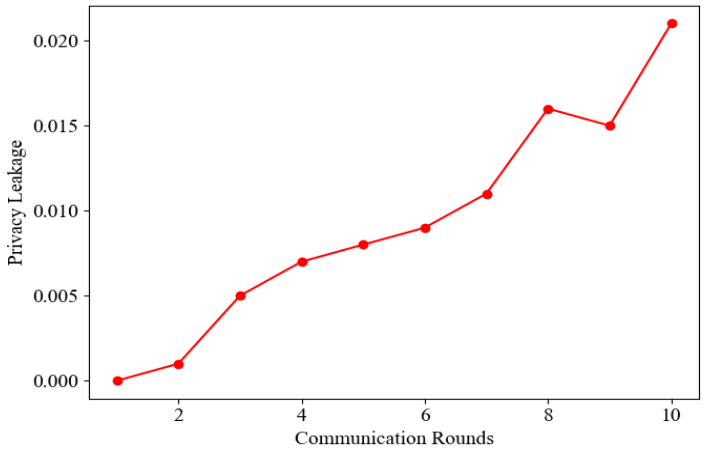

Fig. 6. Privacy leakage of Proposed FL-ODP-DFT Method

5.3.4 Training Time

Training time is typically expressed as the sum of local training times for each client and the communication time for each round. In FL-ODP-DFT, training time is optimized through DFT, which reduces the size of data sent, adaptive clipping, and noise addition based on ODP. These mechanisms reduce the processing time per client, lower

communication overhead, and enhance overall training efficiency. The proposed method demonstrated significantly lower Training time, with a value of 1.5 sec, as shown in Fig.7.

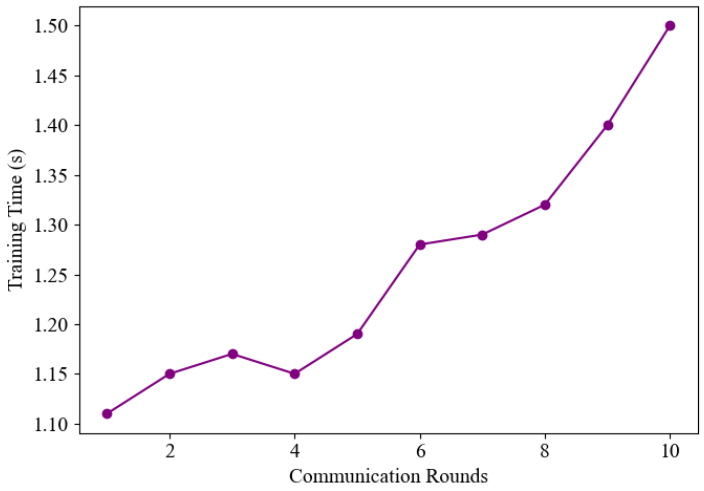

Fig. 7. Training time of Proposed FL-ODP-DFT Method

5.3.5 Computational Efficiency

Computational efficiency can be measured by the complexity of local computation on each client. FL-ODP-DFT enhances computational efficiency by reducing data dimensionality through DFT and minimizing computationally expensive operations via adaptive clipping. The ODP mechanism carefully calibrates noise addition, maintaining model performance without excessive computation. The proposed method demonstrated a notable improvement in computational efficiency, achieving 92.5%, as illustrated in Fig. 8.

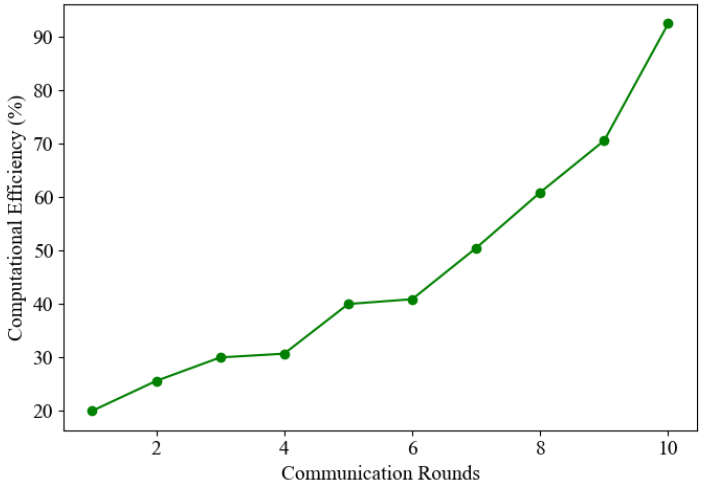

Fig. 8. Computational efficiency of Proposed FL-ODP-DFT Method

5.3.6 Communication Cost

Communication cost is expressed in terms of the data volume transferred between clients and the server over each round. In FL-ODP-DFT, communication cost is minimized by applying DFT to compress gradient data and limiting excessive gradient sizes via clipping. These transformations reduce the data volume exchanged between clients and the server, making FL-ODP-DFT scalable for distributed learning environments. As illustrated in Fig.9, the proposed method achieved a notably reduced communication cost of 100 KB.

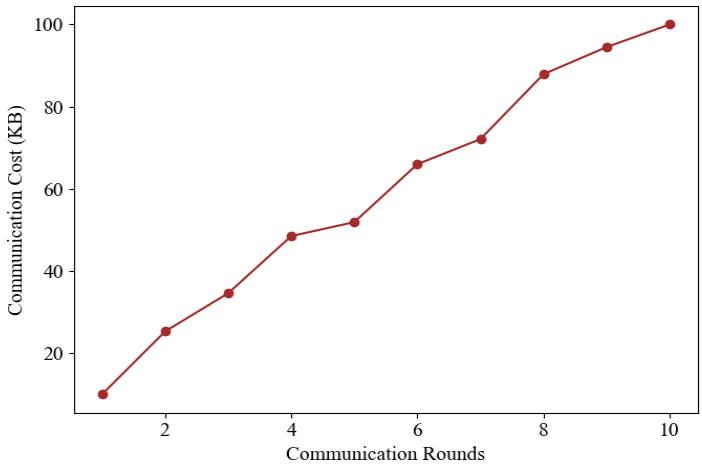

Fig. 9. Communication cost of Proposed FL-ODP-DFT Method

5.4 Comparative Analysis

The model's effectiveness is evaluated by comparing key metrics such as accuracy, training time, privacy budget, privacy leakage, computational efficiency, and communication cost against those of other established models. Moreover, the existing techniques like a FLBM-IoT, PPFLB, FL models [24], zero-concentrated differential privacy (ZcDP), Differentially-Private Stochastic Gradient Descent (DP-SGD), adaptive Gaussian Clipping-DFT (GC-DFT) [28], FL-DP, Local DP-Federated syetem (LDP-Fed) [29].

5.4.1 Accuracy Comparison of Proposed and Existing Methods

In comparing accuracy across different methods as given in Table 4, the proposed FL-ODP-DFT achieves the highest accuracy at 98.9%, indicating superior model performance in FL with optimal privacy and efficiency. It outperforms other models like FL models (97.69%), FLBM-IoT (91.67%), and PPFLB (89.27%). This indicates that FL-ODP-DFT effectively balances model accuracy with enhanced privacy and computational efficiency, making it ideal for privacy-sensitive applications. Fig.10. shows the comparison graph of accuracy.

Table 4. Comparison values of Accuracy

Methods	**Accuracy (%)**
FLBM-IoT	91.67
PPFLB	89.27
FL models	97.69
Proposed FL-ODP-DFT	98.9

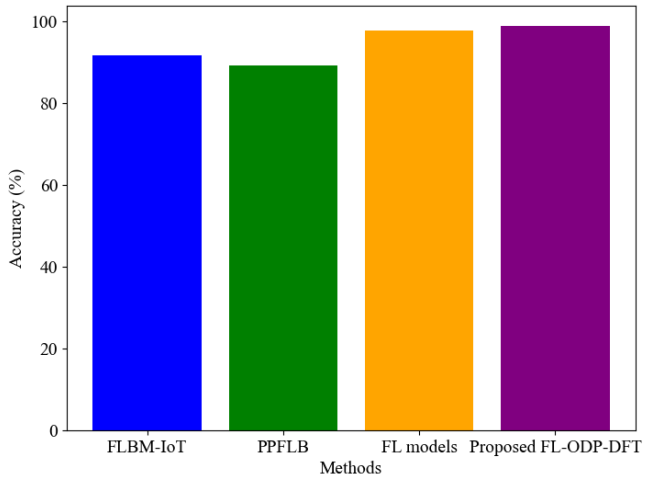

Fig. 10. Comparison graph of Accuracy

5.4.2 Privacy Budget Comparison of Proposed and Existing Methods

The privacy budget comparison shows that the proposed FL-ODP-DFT method offers stronger privacy protection with a lower budget (2.0844) than FL-ODP (3.4) and LDP-Fed (4.1). This lower privacy budget reflects enhanced data

privacy, as FL-ODP-DFT effectively limits privacy leakage by combining optimal differential privacy with DFT, ensuring robust privacy without sacrificing model performance. Fig.11 shows the comparison graph of privacy budget.

Table 5. Comparison values of privacy budget

Methods	**Privacy Budget**
FL-ODP	3.4
LDP-Fed	4.1
Proposed FL-ODP-DFT	2.0844

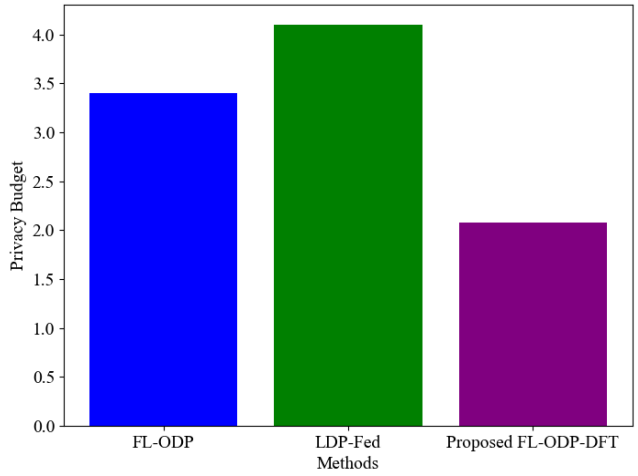

Fig. 11. Comparison graph of privacy budget

5.4.3 Privacy LeakageComparison of Proposed and Existing Methods

In comparing privacy leakage among different models as shown in Table 6, the proposed FL-ODP-DFT model demonstrates the lowest privacy leakage at 0.021, demonstrating its robust capability to safeguard sensitive data throughout the FL process. This represents an improvement over other models, such as general FL models (0.025), PPFLB (0.038), and FLBM-IoT (0.043), each of which has higher privacy leakage values. FLBM-IoT, with the highest leakage at 0.043, is relatively less effective in safeguarding privacy, followed by PPFLB. The reduction in privacy leakage with FL-ODP-DFT highlights its enhanced design for privacy preservation. Fig.12. shows the comparison graph of privacy leakage.

Table 6. Comparison values of Privacy Leakage

Methods	**Privacy Leakage**
PPFLB	0.038
FLBM-IoT	0.043
FL models	0.025
Proposed FL-ODP-DFT	0.021

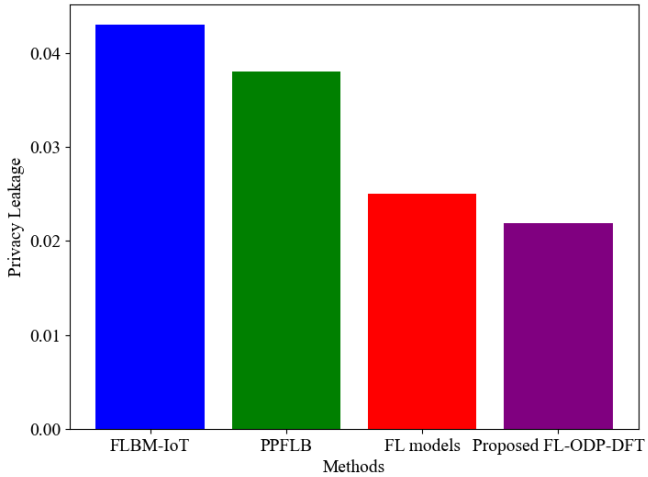

Fig. 12. Comparison graph of privacy leakage

5.4.4 Training Time Comparison of Proposed and Existing Methods

The training time comparison between different methods highlights the significant efficiency of the proposed FL-ODP-DFT approach as given in Table 7. While traditional methods like ZcDP (475 seconds) and DP-SGD (468 seconds) require substantial time for each training round due to extensive privacy-preserving computations and communication overhead, the GC-DFT method improves efficiency with a reduced training time of 72 seconds by leveraging the DFT for data compression. However, the proposed FL-ODP-DFT method drastically improves upon these by achieving a remarkably low training time of just 1.5 seconds. This reduction is attributed to its optimized use of differential privacy with minimal noise addition, efficient client-side computations, making it highly scalable and efficient for FL tasks. Fig.13. shows the comparison graph of training time.

Table 7. Comparison values of Training Time

Methods	**Training Time (s)**
ZcDP	475
DP-SGD	468
GC-DFT	72
Proposed FL-ODP-DFT	1.5

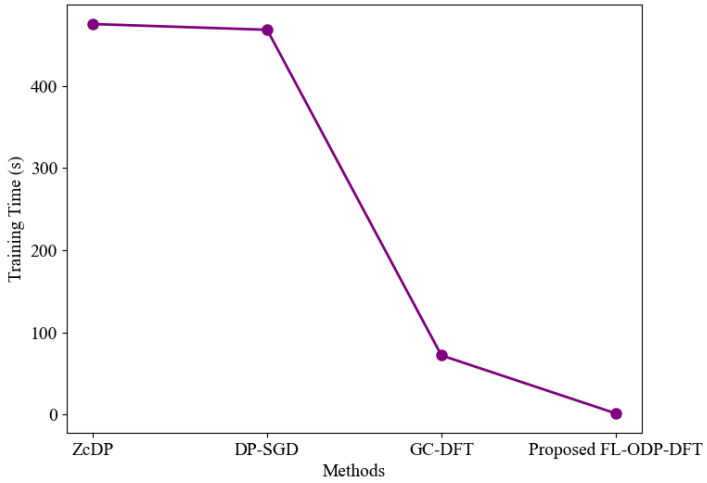

Fig. 13. Comparison graph of Training time

5.4.5 Computational Efficiency Comparison of Proposed and Existing Methods

The computational efficiency comparison given in Table 8 highlights the performance of various FL models, with the Proposed FL-ODP-DFT method achieving the highest efficiency at 92.5%. This indicates that FL-ODP-DFT outperforms other models like FLBM-IoT (70%), PPFLB (75%), and traditional FL models (85%) in terms of reducing resource usage while maintaining high performance. The FL-ODP-DFT method enhances efficiency by optimizing data transformations through DFT, reducing the need for computationally intensive operations, and effectively managing privacy constraints. Fig.14. shows the comparison graph of computational efficiency.

Table 8. Comparison values of Computational Efficiency

Methods	**Computational Efficiency (%)**
FLBM-IoT	70
PPFLB	75
FL models	85
Proposed FL-ODP-DFT	92.5

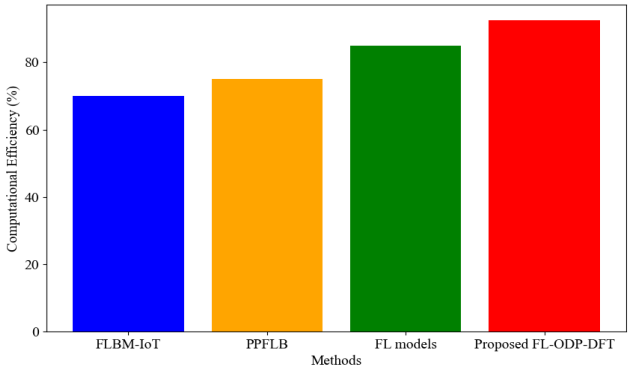

Fig. 14. Comparison graph of computational efficiency

5.4.6 Communication Cost Comparison of Proposed and Existing Methods

The comparison of communication costs given in Table 9 (measured in kilobytes, KB) between different methods highlights the efficiency of the proposed FL-ODP-DFT approach. The proposed FL-ODP-DFT method outperforms existing methods. While ZcDP and DP-SGD have communication costs of 186.88 KB and 184.32 KB respectively, the GC-DFT method reduces it to 133.12 KB by leveraging DFT. The FL-ODP-DFT method further optimizes this, lowering the communication cost to 100 KB, demonstrating its efficiency in minimizing data exchange while maintaining privacy and model performance. Fig.15. shows the comparison graph of communication costs.

Table 9. Comparison values of Communication Cost

Methods	**Communication Cost (KB)**
ZcDP	186.88
DP-SGD	184.32
GC-DFT	133.12
Proposed FL-ODP-DFT	100

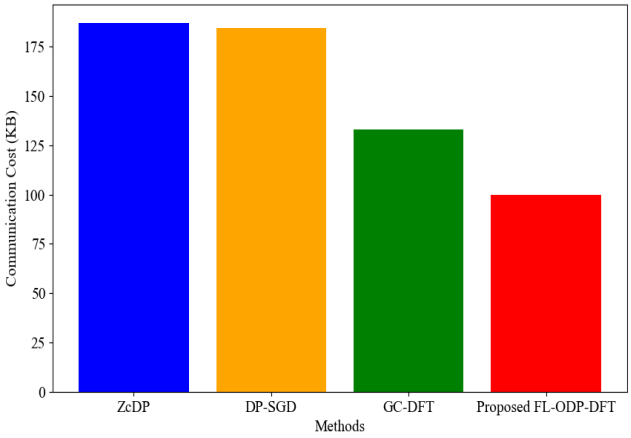

Fig. 15. Comparison graph of communication costs

5.5 Overall Performance of the Proposed

The proposed FL-ODP-DFT method demonstrates strong overall performance in key metrics as shown in Fig.16, highlighting its effectiveness in balancing privacy and model performance in FL. The method achieves an impressive accuracy of 98.9%, ensuring high model performance. It minimizes privacy leakage to a low value of 0.021, showcasing robust privacy protection. With computational efficiency at 92.5%, the method performs optimally without excessive resource consumption. The training time is efficient at just 1.5 seconds per iteration. The privacy budget of 2.0844 ensures privacy preservation while maintaining model accuracy. Finally, the communication cost of 100 KB is kept low, making the method suitable for real-world FL deployments. Overall, the FL-ODP-DFT method effectively balances privacy, accuracy, and efficiency, with excellent results across all evaluated metrics.

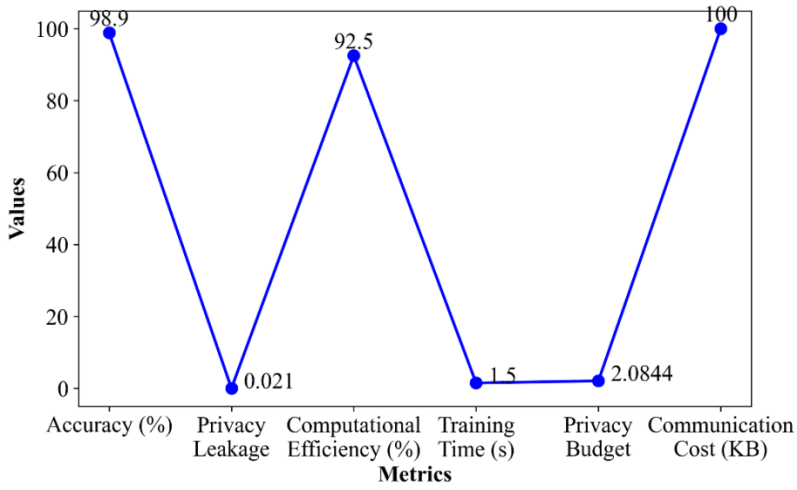

Fig. 16. Overall performance of the proposed FL-ODP-DFT method

The proposed FL-ODP-DFT framework is well-suited for real-world applications in privacy-sensitive domains. In healthcare, where patient data confidentiality is paramount, hospitals can collaboratively train diagnostic models (e.g., for cancer detection or risk prediction) without revealing patient records. The DFT transformation obfuscates sensitive features in gradients, while ODP ensures rigorous differential privacy. In the financial sector, institutions can jointly develop fraud detection models across distributed transaction logs. AGC stabilizes contributions from highly variable datasets, and gradient compression via DFT reduces the burden of transmitting large model updates. For IoT environments, which often involve resource-constrained devices, FL-ODP-DFT enables lightweight communication and efficient local processing, supporting real-time anomaly detection or predictive maintenance without compromising user or sensor-level data privacy. These domain-specific advantages demonstrate that the proposed method is not only theoretically sound but also practically deployable in diverse, high-stakes scenarios where privacy, efficiency, and adaptability are crucial.

5.6 Scalability Considerations

Although the proposed FL-ODP-DFT framework demonstrates strong results with a small number of 5 clients, its architectural components are designed to scale effectively in larger federated systems. The use of DFT provides gradient compression, significantly reducing communication overhead — a critical factor when scaling to hundreds or thousands of clients. Additionally, AGC dynamically adjusts gradient clipping thresholds based on local statistics, which is especially beneficial in heterogeneous client environments common at scale.

The ODP mechanism enables per-client noise calibration based on sensitivity and budget, supporting differential privacy at scale without degrading overall model performance. Since noise addition is performed in the frequency domain, the method also offers natural resistance to privacy attacks even with more participating clients. Moreover, communication is limited to transmitting compressed and noise-protected gradients, reducing server-side congestion during aggregation.

While initial experiments focused on a modest number of clients for controlled evaluation, future work will explore performance under large-scale FL settings (e.g., 100+ clients) to validate convergence behavior, communication efficiency, and privacy leakage under diverse data distributions and varying client participation rates.

Although FL-ODP-DFT includes multiple stages such as AGC, DFT, and ODP noise injection, which may increase processing time, it remains suitable for near-real-time or periodic decision-making scenarios, such as fraud detection and health monitoring. The reduced communication cost from DFT helps offset local computation time. Future work will focus on latency optimization through lightweight DFT approximations and asynchronous updates to enable broader real-time applicability.

6. Conclusion

The proposed FL-ODP-DFT method presents a significant advancement in FL by integrating advanced privacy-preserving strategies with efficient gradient processing techniques. By combining Optimal Differential Privacy with Discrete Fourier Transform and Adaptive Gaussian Clipping, the FL-ODP-DFT approach effectively addresses the challenges of privacy leakage and gradient distortion in FL systems. The DFT transformation reduces gradient size and provides an additional layer of security, while the ODP mechanism optimally adds noise based on the privacy budget, ensuring a balanced trade-off between privacy and model accuracy. Experimental results show that the FL-ODP-DFT method enhances privacy protection while maintaining high model performance, improving computational efficiency, and reducing training time. Future work will focus on scaling FL-ODP-DFT for large federated networks, improving aggregation efficiency, and addressing client heterogeneity. It will also explore dynamic privacy budgets and

integration with other privacy technologies like homomorphic encryption. Additionally, the method's application in privacy-sensitive domains like healthcare and finance will be further investigated.

References

[1] N. Khalid, A. Qayyum, M. Bilal, A. Al-Fuqaha, and J. Qadir, Privacy-preserving artificial intelligence in healthcare: Techniques and applications. *Computers in Biology and Medicine*, vol.*158*, 2023, p.106848.

[2] L. Lyu, H. Yu, X. Ma, C. Chen, L. Sun, J. Zhao, Q. Yang, and S.Y. Philip, Privacy and robustness in federated learning: Attacks and defenses. *IEEE transactions on neural networks and learning systems*. 2022.

[3] M. Zhu, J. Yuan, G. Wang, Z. Xu, and K. Wei, Enhancing Collaborative Machine Learning for Security and Privacy in Federated Learning. *Journal of Theory and Practice of Engineering Science*, vol.*4*, no. 02, 2024, pp.74-82.

[4] O. Aouedi, A. Sacco, K. Piamrat, and G. Marchetto, Handling privacy-sensitive medical data with federated learning: challenges and future directions. *IEEE journal of biomedical and health informatics*, vol.*27*, no. 2, 2022, pp.790-803.

[5] Z. Mahmood, and V. Jusas, Blockchain-enabled: Multi-layered security federated learning platform for preserving data privacy. *Electronics*, vol.*11*, no. 10, 2022, p.1624.

[6] T. Nguyen, and M.T. Thai, Preserving privacy and security in federated learning. *IEEE/ACM Transactions on Networking*. 2023.

[7] N.N. Sakhare, R. Kulkarni, N. Rizvi, D. Raich, A. Dhablia, and S.P. Bendale, A Decentralized Approach to Threat Intelligence using Federated Learning in Privacy-Preserving Cyber Security. *Journal of Electrical Systems*, vol.*19*, no. 3, 2023.

[8] J. Liu, J. Huang, Y. Zhou, X. Li, S. Ji, H. Xiong, and D. Dou, From distributed machine learning to federated learning: A survey. *Knowledge and Information Systems*, vol.*64*, no. 4, 2022, pp.885-917.

[9] M. Butt, N. Tariq, M. Ashraf, H.S. Alsagri, S.A. Moqurrab, H.A.A. Alhakbani, and Y.A. Alduraywish, A Fog-Based Privacy-Preserving Federated Learning System for Smart Healthcare Applications. *Electronics*, vol.*12*, no. 19, 2023, p.4074.

[10] S. Dasari, and R. Kaluri, 2P3FL: A Novel Approach for Privacy Preserving in Financial Sectors Using Flower Federated Learning. *CMES-Computer Modeling in Engineering and Sciences*, vol.*140*, no. 2, 2024, pp.2035-2051.

[11] J. Zhang, J. Zhang, D.W.K. Ng, and B. Ai, Federated learning-based cell-free massive MIMO system for privacy-preserving. *IEEE Transactions on Wireless Communications*, vol.*22*, no. 7, 2022, pp.4449-4460.

[12] A. Munawar, and M. Piantanakulchai, A collaborative privacy-preserving approach for passenger demand forecasting of autonomous taxis empowered by federated learning in smart cities. *Scientific Reports*, vol.*14*, no. 1, 2024, p.2046.

[13] M.F. Almufareh, N. Tariq, M. Humayun, and B. Almas, A federated learning approach to breast cancer prediction in a collaborative learning framework. In *Healthcare* Vol. 11, No. 24, 2023, p. 3185. MDPI.

[14] Baabdullah, T. Alzahrani, A. Rawat, D.B. and Liu, C. 2024. Efficiency of Federated Learning and Blockchain in Preserving Privacy and Enhancing the Performance of Credit Card Fraud Detection (CCFD) Systems. *Future Internet*, vol.*16*, no. 6, p.196.

[15] Q. Wang, H. Yin, T. Chen, J. Yu, A. Zhou, and X. Zhang, Fast-adapting and privacy-preserving federated recommender system. *The VLDB Journal*, vol.*31*, no. 5, 2022, pp.877-896.

[16] F. Yu, Z. Xu, Z. Qin, and X. Chen, Privacy-preserving federated learning for transportation mode prediction based on personal mobility data. *High-Confidence Computing*, vol.*2*, no. 4, 2022, p.100082.

[17] V. Michalakopoulos, E. Sarantinopoulos, E. Sarmas, and V. Marinakis, Empowering federated learning techniques for privacy-preserving PV forecasting. *Energy Reports*, vol.*12*, 2024, pp.2244-2256.

[18] Z. Chen, J. Li, L. Cheng, and X. Liu, Federated-WDCGAN: A federated smart meter data sharing framework for privacy preservation. *Applied Energy*, vol.*334*, 2023, p.120711.

[19] Z. Liu, J. Guo, W. Yang, J. Fan, K.Y. Lam, and J. Zhao, Dynamic user clustering for efficient and privacy-preserving federated learning. *IEEE Transactions on Dependable and Secure Computing*. 2024.

[20] Y. Wang, A. Zhang, S. Wu, and S. Yu, VOSA: Verifiable and oblivious secure aggregation for privacy-preserving federated learning. *IEEE Transactions on Dependable and Secure Computing*, vol.*20*, no. 5, 2022, pp.3601-3616.

[21] J. Song, W. Wang, T.R. Gadekallu, J. Cao, and Y. Liu, Eppda: An efficient privacy-preserving data aggregation federated learning scheme. *IEEE Transactions on Network Science and Engineering*, vol.*10*, no. 5, 2022, pp.3047-3057.

[22] T. Eltaras, F. Sabry, W. Labda, K. Alzoubi, and Q. Ahmedeltaras, Efficient verifiable protocol for privacy-preserving aggregation in federated learning. *IEEE Transactions on Information Forensics and Security*, vol.*18*, 2023, pp.2977-2990.

[23] J. Ma, S.A. Naas, S. Sigg, and X. Lyu, Privacy-preserving federated learning based on multi-key homomorphic encryption. *International Journal of Intelligent Systems*, vol.*37*, no. 9, 2022, pp.5880-5901.

[24] M. Abaoud, M.A. Almuqrin, and M.F. Khan, Advancing federated learning through novel mechanism for privacy preservation in healthcare applications. *IEEE Access*, *11*, 2023, pp.83562-83579.

[25] X. Wu, L. Xu, and L. Zhu, Local Differential Privacy-Based Federated Learning under Personalized Settings. *Applied Sciences*, vol.*13*, no. 7, 2023, p.4168.

[26] A. Yazdinejad, A. Dehghantanha, H. Karimipour, G. Srivastava, and R.M. Parizi, A robust privacy-preserving federated learning model against model poisoning attacks. *IEEE Transactions on Information Forensics and Security*. 2024.

[27] M. Abdel-Basset, H. Hawash, N. Moustafa, I. Razzak, and M. Abd Elfattah, Privacy-preserved learning from non-iid data in fog-assisted IoT: A federated learning approach. *Digital Communications and Networks*. 2022.

[28] M.A. Hidayat, Y. Nakamura, and Y. Arakawa, Enhancing Efficiency in Privacy-preserving Federated Learning for Healthcare: Adaptive Gaussian Clipping with DFT Aggregator. *IEEE Access*. 2024.

[29] M. Iqbal, A. Tariq, M. Adnan, I.U. Din, and T. Qayyum, FL-ODP: An optimized differential privacy enabled privacy preserving federated learning. *IEEE Access*, vol. *11*, 2023, pp.116674-116683.

[30] R. Madduri, Z. Li, T. Nandi, K. Kim, M. Ryu, and A. Rodriguez, Advances in Privacy Preserving Federated Learning to Realize a Truly Learning Healthcare System. *arXiv preprint arXiv:2409.19756*. 2024.

[31] K.A. Awan, I.U. Din, A. Almogren, and J.J. Rodrigues, Privacy-Preserving Big Data Security for IoT With Federated Learning and Cryptography. *IEEE Access*, vol. *11*, 2023, pp.120918-120934.

[32] J. Xiao, X. Tang, and S. Lu, Privacy-Preserving Federated Class-Incremental Learning. *IEEE Transactions on Machine Learning in Communications and Networking*.2023.
[33] Z. Zhang, X. Ma, and J. Ma, Local Differential Privacy Based Membership-Privacy-Preserving Federated Learning for Deep-Learning-Driven Remote Sensing. *Remote Sensing*, vol. *15*, no. 20, 2023, p.5050.
[34] Y. Wang, Z. Su, Y. Pan, T.H. Luan, R. Li, and S. Yu, Social-aware clustered federated learning with customized privacy preservation. *IEEE/ACM Transactions on Networking*. 2024.
[35] Y. Huang, Y. Song, and Z. Jing, Load forecasting using federated learning with considering electricity data privacy preservation of EASP. *Ain Shams Engineering Journal*, vol.*15*, no. 6, 2024, p.102724.

Authors' Profiles

Deny P. Francis is currently working as an Assistant Professor in the Department of Computer Science at Naipunnya Institute of Management and Information Technology, Thrissur, Kerala, 680308, India. His research interests are in the area of Artificial Intelligence and Machine learning. He is a member of professional bodies such as Computer Society of India.

Sharmila R. is currently working as an Professor in the Department of Computer Science(Applications) at Karpagam Deemed to be university, Coimbatore - 641 021, Tamil Nadu, India. Their research interests are in the area of Computer applications, AI and ML.

How to cite this paper: Deny P. Francis, R. Sharmila, "Innovative Privacy Preserving Strategies in Federal Learning", International Journal of Information Engineering and Electronic Business(IJIEEB), Vol.17, No.6, pp. 112-133, 2025. DOI:10.5815/ijieeb.2025.06.09