Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Double Masking as a Dual-Lock Security for Privacy-Preserving in Federated Learning

Gehad Elsaady

jihad_pg24@ai.kfs.edu.eg

Kafrelsheikh University

Zainab Ali

Kafrelsheikh University

Mahmoud Shams

Kafrelsheikh University

Article

Keywords: Federated learning, privacy preserving, double masking, privacy leakage, gradient selection

Posted Date: November 20th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11419924/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Double Masking as a Dual-Lock Security for Privacy-Preserving in Federated Learning

Gehad E. Elsaady [a], Zainab H. Ali [b,c], and Mahmoud Y. Shams [d]

[a] Department of Machine learning and Information Retrieval, Faculty of Artificial Intelligence, Kafrelsheikh University, El-Geish st, Kafrelsheikh, 33516, Egypt. (corresponding author)
E-mail: jihad_pg24@ai.kfs.edu.eg.

[b] Department of Embedded Network Systems and Technology, Faculty of Artificial Intelligence, Kafrelsheikh University, El-Geish st, Kafrelsheikh, 33516, Egypt, [c] Department of Electronics and Computer Engineering, School of Engineering and Applied Sciences at Nile University, Giza, Egypt.

[d] Department of Machine learning and Information Retrieval, Faculty of Artificial Intelligence, Kafrelsheikh University, El-Geish st, Kafrelsheikh, 33516, Egypt.

Abstract

As awareness of data privacy increases and data protection requirements become more inflexible, federated learning (FL) has emerged as a viable solution that facilitates collaborative model training without the need to share raw data with a central server. Despite its advantages, FL remains vulnerable to privacy leakage through shared gradients and suffers from high computational overhead and communication costs. To this end, this paper introduces a novel double privacy masking framework for FL, which integrates both a Pseudo-Random Number Generator (PRNG)-based client-side masking mechanism with a hash-based server-side masking approach. To further enhance efficiency, the proposed framework incorporates a gradient selection strategy along with the Adam optimizer. The combination of PRNG and hash-based masks effectively prevents the central server from working out sensitive information from client-uploaded gradients, significantly strengthening privacy protection. Experimental results demonstrate that the proposed approach reduces privacy leakage by 68% while preserving model accuracy at 99.13%. Moreover, the integration of gradient selection and Adam optimization decreases computational overhead by 7.25% without increasing communication traffic. These results indicate that the proposed privacy-preserving FL framework achieves a profitable balance between privacy, efficiency, and accuracy, making it particularly suitable for resource-constrained devices and edge computing environments.

Keywords

Federated learning; privacy preserving; double masking; privacy leakage; gradient selection

1. Introduction

Over recent years, Machine Learning (ML) has evolved quickly and has become a

fundamental tool in various academic research areas [1]. ML can efficiently complete an assortment of challenging tasks [2]. The magnitude of data continues to grow and awareness of privacy increases, traditional centralized ML methods are facing greater challenges [3]. laws have been established to protect data like the Cybersecurity Law of the People's Republic of China in 2017, General Data Protection Regulation (GDPR) passed by the European Union in 2018, and the California Consumer Privacy Act (CCPA) in the United States in 2020. As a result, data privacy protection has emerged as a major research priority in the ML community [4]. Unlike in the past, personal data is no longer easily accessible, which increases the need for new learning techniques that respect users' privacy while still enabling effective model training [5].

FL, is a distributed ML approach, proficiently addressing problems related to data privacy and centralization by training models on local servers and aggregating the updated model parameters [7]. FL is introduced as a distributed, privacy-preserving ML architecture. FL allows multiple parties to collaboratively train models without sharing their local private data with the central server. By sharing the trained models, they achieve a training outcome that is equivalent to what would have been attained if the private data had been shared directly. This approach significantly mitigates the risk of data privacy breaches while simultaneously lowering training costs across diverse institutions [8][9].

The privacy preserving performance of FL has been enhanced and has been implemented in many areas. A multi-source data privacy protection method combined between homomorphic encryption and blockchain to solve the difficult problem of privacy protection for heterogeneous data in media communication, significantly shortens the processing time, and introduced a comparison between k-anonymity and differential privacy methods, it demonstrates lower encryption and decryption time consumption and widespread application potential[10]. Introducing privacy-preserving and communication-efficient smart grid energy prediction scheme based on FL, applying a deep learning model combined Long Short-Term Memory networks (LSTM) and Convolutional Neural Networks (CNN) to forecast future changes in grid energy while ensuring the security of user data [11].

A privacy-aware asynchronous FL framework based on peer-to-peer networking for secure and resilient decentralized model training in modern mobile robotic systems within 5G and beyond networks. This Model Consists of a reputation-aware coordination mechanism, a communication mechanism based on secret sharing, and a secure stochastic gradient descent (SGD) scheme, which can robustly enhance security [12]. A privacy-preserving FL model is proposed to integrate internal audit, additive homomorphic encryption, and Gaussian mixture models. This model defends against poisoning attacks and optimizes computational and communication costs [13]. A novel training paradigm called Secure Federated Learning (SFL) is proposed, which Integrated watermarks into the parameters of deep neural network models to protect the intellectual property of FL models, aiming to solve the problems of data privacy and model intellectual property protection [14].

To enhance the privacy protection performance and reduce privacy leakage in FL systems, by combining differential privacy with effective clipping and the Gaussian mechanism was acritical strategy to preserving data, it ensures that malicious participants or curious servers cannot get any private information from the uploaded data and achieved high accuracy with low differential [15]. Even with these studies that have enhanced the privacy

preservation performance of FL systems, there is a need to decrease privacy leakage and increase accuracy.

Based on the previous research approaches, to enhance privacy preserving performance, increase accuracy and reduce communication overhead in FL systems, a new privacy preservation method is introduced and applied to FL systems. The contributions of this research are as follows:

I. Enhance privacy preserving performance by using an improved double privacy masking protocol using PRNG and Hash Based mask, it guarantees that adversarial participants or inquisitive servers are unable to access or infer any sensitive information from the uploaded data.

II. Reduce the value of privacy leakage by 68% by calculating the average norm of the gradients before and after applying masks.

III. Increase accuracy from 96.46% in recent studies to 99.13% in this FL model by using ADAM optimizer.

IV. Increased computational efficiency by employing TRQ (Top-Random-Q) gradient selection algorithm, the volume of data that needs to be encrypted is significantly reduced, thereby drastically decreasing the time required for computation.

V. Maintaining the stability of communication traffic to be the same value before and after using masks

VI. Decrease computational overhead by 7.25% after using double masking technique.

The structure of this study is as follows: Section 2 reviews background and related works about FL and its definition, types and techniques. Section 3 introduces privacy and preserving FL via double masking technique, inside this section the value of double masking and other algorithms that are employed in the proposed approach is highlighted. Section 4 describes the methodology and methods. Section 5 explains the results and discussion. Section 6 provides a systematic summary of the paper as a conclusion.

2. Related Works and Background

To increase privacy preserving in FL systems, researchers have continually introduced new techniques since FL first emerged. Chen et al. [16] implemented a privacy preserving and traceable FL framework by enhancing the aggregation architecture and integrating blockchain with the Interplanetary File System (IPFS), the framework enables secure data sharing and model traceability in industrial IoT settings. Similarly, Wang et al. [17] introduced a privacy-preserving FL (PPFL) framework for a user-grouping strategy designed to balance time consumption and the number of training rounds in intelligent medical applications. Their model results show that both the size of the public key set and the number of users involved in training significantly influence signing and verification time, while the framework continued to preserve strong privacy guarantees and efficient performance. Zhang et al. [15] enhance the privacy protection performance and reduce the privacy leakage in FL systems by combining differential privacy (DP) with effective clipping and the Gaussian mechanism it ensures that malicious participants or curious servers cannot get any private information

from the uploaded data and achieved high accuracy 96.46% with differential privacy 2.0 so that is considered a significant achievement. All this research introduced new steps for enhancement privacy preserving in Fl but still require improvement to increase accuracy and ensure that there is no privacy leakage so that this paper will do. Song et al. [18] proposed an Efficient Privacy Preserving Data Aggregation (EPPDA) scheme utilizes encryption and data aggregation methods to mitigate privacy leakage risks in FL.

On the other hand, Namakshenas et al. [19] proposed FL approach based on quantum center verification and additive homomorphic encryption. Experiments across multiple datasets demonstrated that their scheme is both accurate and efficient, offering notable improvements to privacy and security for consumer IoT systems. Despite these advances, traditional FL still suffers from challenges such as high communication overhead and insufficient privacy protection during model aggregation, highlighting the need for further optimization. Ma et al. [20] addressed these issues by proposing a PPFL scheme based on multi-key homomorphic encryption (HE). Their work identifies major privacy risks in FL, reviews current protection techniques, and introduces a solution capable of preventing information leakage and device collusion. In another contribution, Zhang et al. [21] examined the use of HE-based PPFL in IoT healthcare environments and presented a framework that supports distributed model training and updates while protecting the privacy of user data. Nguyen et al. [22] proposed a framework that integrates secure aggregation with client-side defense mechanisms to counter malicious user behaviors. By assigning responsibility for defense actions to users and verifying their correctness through zero-knowledge proof protocols, the framework unifies secure aggregation with robust defense strategies. Shwkat et al. [6] proposed RMPFD framework improves the convergence rates and classification accuracy, leading to an improvement of over 12% in performance compared to traditional FL methods. Rahmati proposed FL model that achieved 91.2% model accuracy, improves task success rates by 7.6% and reduces privacy leakage risks by 41.5% compared to centralized AI models [79]. Related work from recent researches is shown in Table 1.

Table 1: Related work

reference	Key Technique(s)	Key Results & Metrics
Song et al. (2022) [18]	EPPDA: Encryption & Data Aggregation	Efficiently mitigated leakage risks in standard aggregation.
Zhang et al. (2023) [21]	HE-based Privacy Protection	Protected user data in distributed healthcare updates.
Chen et al. (2024) [16]	Blockchain + IPFS Integration	Secured data sharing and enabled model traceability.
Namakshenas(2024) [19]	Quantum Verification + Additive HE	Improved security against quantum-level threats in IoT.
Ma et al. (2024) [20]	Multi-key Homomorphic Encryption	Prevented device collusion and information leakage.

Nguyen et al. (2024) [22]	Secure Aggregation + ZK Proofs	Defended against malicious behavior via client-side verification.
Zhang et al. (2025) [15]	DP + Gaussian Mechanism + Clipping	96.46% Accuracy
Wang et al.(2025) [17]	User-grouping Strategy	Optimized time/rounds; preserved strong privacy.
Shwkat et al.(2025) [6]	RMPFD Framework	>12% performance boost in accuracy/convergence.
Rahmati (2025) [79]	Privacy Leakage Prevention Model	91.2% Accuracy; 41.5% reduction in leakage risks.

2.1.FL Architecture and Operational Paradigm

FL is a distributed ML paradigm in which multiple decentralized clients collaboratively train a shared global model while ensuring that raw data remains localized at the data source, as illustrated in Figure 1 [23] [24]. This paradigm enables privacy-preserving artificial intelligence by eliminating the need for direct data sharing with a centralized server.

FL is specifically designed to operate within multi-layer cloud–edge–end architectures, making it well-suited for the requirements of modern distributed computing environments such as 5G and emerging 6G networks, Internet of Things ecosystems, and intelligent edge-enabled applications. Unlike traditional centralized cloud-based learning approaches, which are vulnerable to privacy breaches and incur significant communication overhead, FL distributes computational and learning responsibilities across multiple system layers. In this collaborative framework, the cloud layer is responsible for coordinating global model aggregation and long-term model storage, while the edge layer facilitates intermediate aggregation, supports latency-sensitive services, and reduces communication bottlenecks. Meanwhile, end devices, including IoT sensors, mobile terminals, and wearable systems, perform local data collection and compute local model updates. This hierarchical distribution of tasks enhances scalability, reduces network congestion, and strengthens privacy preservation, making FL a key enabling technology for next-generation intelligent systems.

Despite its advantages, FL encounters several challenges, including privacy and security threats such as information leakage and adversarial attacks, heterogeneity in data distribution and device capabilities, and limited computational and communication resources at the edge and end-device levels. Furthermore, the integration of trustworthy AI principles encompassing fairness, robustness, and explainability introduces additional complexity. Owing to its ability to enable learning from decentralized data sources, FL has gained increasing attention in the field of cyber-physical systems (CPS), where tight integration between computation, communication, and physical processes is essential [25][26].

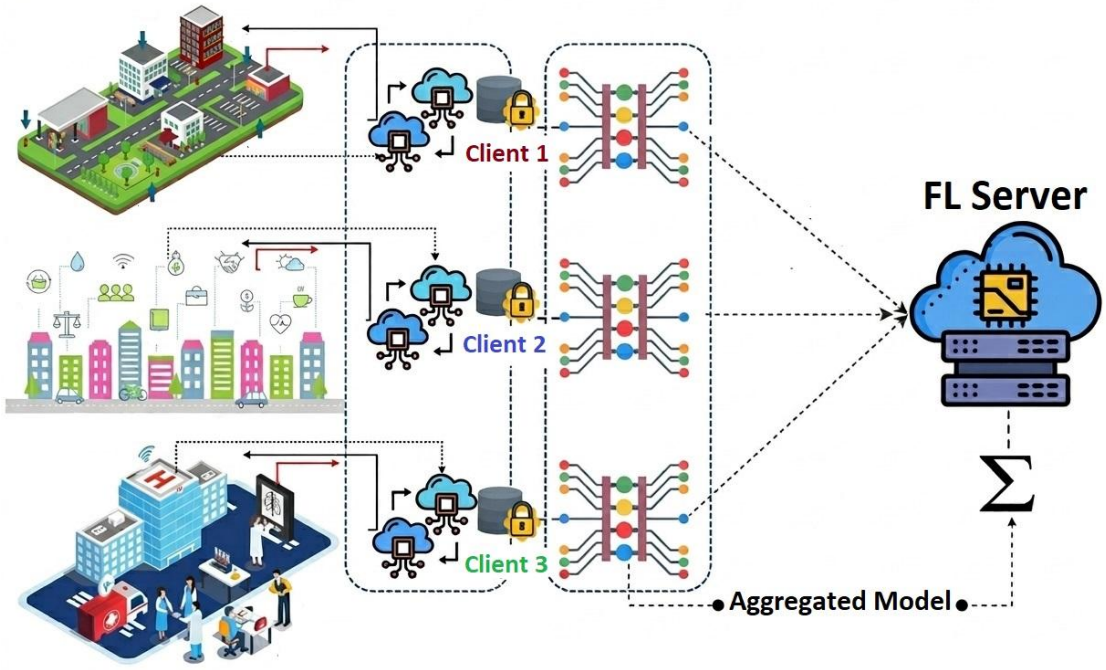


Figure 1: FL concept and approach

2.2.FL Classification and Categorization

FL can be categorized in multiple ways; however, this study adopts a clear and practical classification by organizing FL into seven distinct categories. One of the most common types is Horizontal Federated Learning (HFL), that applies when participating clients share the same feature space but differ in the data samples they possess. In this setting, all clients maintain identical data attributes such as common fields in a database while each holds unique records. A representative example is the healthcare domain, multiple hospitals collect the same types of patient information (e.g., age, diagnosis, and treatment), yet each institution manages data from different patients. HFL enables the collaborative training of a unified global model that leverages the collective knowledge of all participants without requiring any hospital to disclose its sensitive patient data, thereby preserving privacy [27][28].

Vertical Federated Learning (VFL) refers to a FL setting in which participating entities hold different feature spaces for the same set of users or samples, resulting in vertically partitioned data. In such cases, each party contributes complementary information rather than overlapping records. For example, a bank and an e-commerce platform may serve an identical customer base; however, the bank typically maintains financial and transaction-related attributes, whereas the e-commerce platform records users' browsing patterns and purchasing behavior. VFL enables these

parties to collaboratively train a predictive model by integrating heterogeneous feature sets, while ensuring that raw data remains localized and private. VFL is particularly valuable in scenarios where regulatory, ethical, or competitive constraints prohibit direct data sharing or centralized data fusion [27][30].

Federated Transfer Learning (FTL) is a specialized collaborative framework; it is designed to use where organizations possess distinct data samples and non-overlapping feature sets yet share a common analytical goal. Parties can build a sophisticated predictive model without the need to exchange sensitive raw data in FTL instead of the process relies on the transmission of model parameters or gradients, ensuring that the underlying data remains secure behind each participant's firewall. An example of this is the collaboration between a bank and a credit card company. Bank maintains records of loan applications and repayment histories; the credit card company holds insights into daily consumer spending habits. Although their datasets contain different attributes, both entities are interested in the shared objective of predicting credit risk. Through FTL, they can synthesize their unique knowledge to train a single, robust model that captures a more comprehensive view of financial behavior than could achieve in isolation, all while maintaining strict data privacy [27][30].

Decentralized Federated Learning (DFL) gets rid of the middleman the central server. This is a peer-to-peer approach to FL where clients communicate and exchange model updates directly with each other, rather than having a single entity aggregate and distribute everything. Because of this structure, DFL is a natural fit for systems built on blockchain or other distributed ledger technologies, perfectly aligned with their decentralized and trustless design [29-31].

Federated Multi-Task Learning (FMTL) is a smart approach; every client gets to train its own personalized model while simultaneously sharing valuable knowledge with a global model. This structure gives all participants the best of both worlds: they benefit from the collective data of the entire network without being forced to use a generic, one-size-fits-all solution. FMTL is especially effective when the data held by different clients varies significantly, allowing it to create much more accurate, customized models for everyone involved [34][35].

Federated Reinforcement Learning (FRL) brings together FL and reinforcement learning, allowing multiple independent agents to work together and collaboratively learn a shared set of rules for sequential decision-making tasks without ever sharing their raw experience data. This approach is highly practical for fields like robotics or autonomous driving, where a central model can be trained on the combined experiences of many devices. This collective knowledge improves the overall performance and safety of the system while ensuring the privacy of each individual device's operational data is strictly maintained [38]. Hybrid Federated Learning: in this case the FL can combine different methods to act with complex systems that a single approach can't solve alone. It allows the model to leverage the strengths of different techniques, such as HFL, VFL, or FTL, to handle diverse data distributions, varying client capabilities, and multiple learning

objectives within one integrated framework. This approach is highly valuable for intricate, real-world scenarios where data is partitioned in numerous ways across an organization or network. Blockchain and Double masking technique are great examples combined with FL [23,27,35,37].

2.3.FL Optimization and Aggregation Algorithms

Several algorithms have been proposed in FL to address the inherent challenges of decentralized training over heterogeneous and non- Independent and Identically Distributed (non-IID) data. Federated Averaging (FedAvg) [39] is the most adopted baseline approach, where local model updates are aggregated at the server using weighted averaging. Even though FedAvg is efficient and simple structure, it often suffers from slow convergence and performance degradation in highly heterogeneous data settings. To mitigate these issues, Federated Proximal (FedProx) [40] extends FedAvg by introducing a proximal regularization term that stabilizes local updates and improves robustness under system and data heterogeneity. Another effective solution to reduce client drift is SCAFFOLD [41][42], which controls leverages variates to correct local update directions and significantly accelerates convergence in non-IID environments. The FedNova algorithm [44][45] addresses objective inconsistency caused by variable local computation by normalizing client updates before aggregation. In contrast, FedOpt [46][47] incorporates adaptive optimization techniques such as FedAdam and FedYogi, at the server side to enhance convergence speed and training stability.

In addition to global optimization strategies, several algorithms focus on personalization to better accommodate client-specific data distributions. Approaches such as FedPer [50][51] and FedBN [48][49] enable partial model customization, with FedBN demonstrating effectiveness in handling domain shifts by maintaining client-specific batch normalization parameters. More advanced aggregation techniques, including FedMA [43], align neural network layers across clients prior to aggregation, thereby improving model compatibility and performance. Collectively, these FL algorithms address critical challenges in real-world FL deployments, including communication efficiency, robustness to heterogeneity, fairness among clients, and model personalization. A comparative analysis of these methods is illustrated in Table 2.

Table 2: FL Algorithms

Algorithm	Idea	Strengths
FedAvg [39]	Server aggregates local model updates via weighted average	Simple, efficient, widely used baseline
FedProx [40]	Adds a proximal term to local objectives	Stabilizes training, handles heterogeneity
SCAFFOLD [41][42]	Uses control variants to reduce client drift	Effective with non-IID data, improves accuracy
FedMA [44]	Matches neurons across local models before aggregation	Works with deep and structurally different models

FedNova [44][45]	Normalizes local updates to fix objective inconsistency	Reduces bias from uneven training
FedOpt [46][47]	Applies adaptive optimizers at the server	Faster convergence, robust with non-IID data
FedBN [48][49]	Keeps BatchNorm layers local while aggregating others	Handles domain shift well
FedPer [50][51]	Splits model into shared base layers + client-specific layers	Supports personalization for diverse users

3. Privacy-Preserving FL via Double Masking

Double Masking technique is a fundamental security mechanism in FL that enables a central server to aggregate client model updates without ever accessing individual client data or explicit model parameters [52-56] as shown in figure 2. The key objective of this technique is to ensure that privacy is preserved throughout the learning process, even if the server itself is not fully trusted. The technique is referred to as double masking because two independent random masks are applied to each client's model update before it is transmitted. These masks are carefully constructed to protect the update at different stages of the FL workflow. After completing local training, each client completes its model update vector W_i. To prevent the exposure of sensitive information, the client applies two masking components, resulting in the following protected update as shown in Eq. 1:[52]

$$\boldsymbol{W_i^{masked} = W_i + M_i^{client} + M_i^{server}} \quad (1)$$

where M_i^{client}denotes the client-side mask, and M_i^{server}represents the server-related mask component. The client-side mask is generated locally using a pseudo-random process and ensures that the raw update $W_i$is fully obfuscated before leaving the client device. This prevents the server from performing gradient inspection or reconstruction attacks on individual updates. The server-related mask is designed such that its effect is only neutralized when updates from enough clients are aggregated. Importantly, no single masked update can be unmasked in isolation. Masks were structured to be cancelled out only during aggregation, ensuring that the server can recover only the sum of updates, not the individual contributions. Client masks commonly referred to as the Pairwise Mask represent the first and most important line of defense against privacy leakage from both an honest but curious server and other participating clients.

This mechanism is based on a zero-sum mathematical principle, which guarantees that individual model updates remain hidden while the aggregated result remains accurate by requiring every pair of clients $(i, j)$to agree on a shared random seed. Using this seed, client $i$adds a random value $r_{i,j}$to its local model update, while client $j$adds the corresponding negative value $-r_{i,j}$to its own update. As a result, when the central server aggregates all received updates, these pairwise masks naturally cancel each other out $(\cdots + r_{i,j} - r_{i,j} + \cdots)$, allowing the server to recover only the true sum of

model updates without ever observing any individual unmasked contribution. While the pairwise client mask effectively protects privacy during normal operation, real-world FL systems must also account for client dropouts, which can disrupt this cancellation mechanism. When a client disconnects before completing the protocol, the pairwise masks it shares with other clients would remain unbalanced, introducing residual noise into the aggregated model. Mathematical part for traditional double making is shown in Eqs. 2 and 3 [52]. Each client prepares a masked version of its local update vector to ensure privacy before transmission. Specifically, for client i, the masked update $Y_i$is computed as shown in Eq. 2.

$$\boldsymbol{Y_i = W_i + \sum_{j\in U, i<j} s_{i,j} - \sum_{j\in U, i>j} s_{j,i} + b_i (\mathbf{mod} R)} \qquad (2)$$

where $W_i$denotes the raw local model update (gradients or weights), $s_{i,j}$represents the pairwise mask agreed upon between client $i$and client j, $b_i$is the individual mask generated by client $i$to protect its update in case of partner dropouts, and $R$is the modulus used to perform calculations in a finite field to prevent overflow. Pairwise masks ensure intransit confidentiality, while the individual masks provide an additional layer of security against potential inconsistencies arising from client dropout.

Central server receives updates from the set of surviving clients U, the server aggregates the masked updates by computing as shown in Eq. 3.

$$\boldsymbol{\sum_{i\in U} Y_i = \sum_{i\in U} W_i + \sum_{i\in U} \underbrace{\left(\sum_{j\in U, i<j} s_{i,j} - \sum_{j\in U, i>j} s_{j,i} \right)}_{\text{Term A}} + \underbrace{\sum_{i\in U} b_i}_{\text{Term B}}} \qquad (3)$$

A corresponds to the pairwise masks. For each pair $(i, j)$of active clients, $s_{i,j}$is added once by client $i$and subtracted once by client j. Consequently, the sum of all pairwise masks between active clients equals zero, ensuring that the aggregated update reflects only the true local contributions. B represents the individual masks, which cannot be removed directly by the server; instead, the server must request secret shares from the clients to reconstruct $\sum b_i$and subtract it only after confirming that all pairwise masks have canceled out successfully. This mechanism guarantees that the aggregated updates remain confidential throughout the entire process. Even under an honest-but-curious server model, where the server follows the protocol but attempts to infer client data from intermediate values, the updates are never exposed in plaintext at any point. So, the scheme preserves the integrity of the global model while providing robust privacy protection, combining in-transit and at-rest security in a multi-layered masking framework [40-43] but failed to safeguard the model during transmission, leaving the updates susceptible to potential attacks by external by third parties.

This study introduced a new concept based on FL model and double masking technique that called a Dual-Lock Security Architecture to enforce privacy protection through two logically independent

masking layers operating at different stages of the FL workflow as shown in figure 3. The first layer is implemented at the client side, where pseudo-random masks are applied to local model updates prior to transmission. This mechanism guarantees in-transit confidentiality by ensuring that the central server receives only masked gradients and prevents direct inspection or gradient inversion attacks during communication. The correctness of aggregation is preserved through the mathematical cancellation of pairwise masks, which occurs only after summation across all participating clients.

To reinforce client-side protection and address vulnerabilities within the server execution environment, a server-side masking mechanism is introduced as a second security layer. Server mask provides at-rest protection by ensuring that intermediate aggregated updates stored in the server's volatile memory remain masked throughout the aggregation process. Consequently, even under an honest-but-curious threat model where the server faithfully follows the protocol but attempts to infer sensitive information the aggregated gradients are never exposed in plaintext form during intermediate computation stages. The server is only able to recover the usable global update through an explicit and controlled unmasking operation.

Combination between client-side and server-side masking enforces temporal and spatial isolation of sensitive information, significantly reducing the attack surface associated with intermediate gradient exposure and the independence of the two masking layers strengthens the overall security posture of the system. When client masks rely on shared pseudo-random seeds between clients, server masks are generated deterministically using a cryptographic hash known only to server. By this separation ensures that compromising one layer does not invalidate the other and prevents a single point of failure. Dual-Lock Security Architecture extends privacy guarantees beyond data transmission to encompass server-side computation, establishing a defense-in-depth strategy that enhances trustworthiness and robustness in privacy-preserving FL systems.

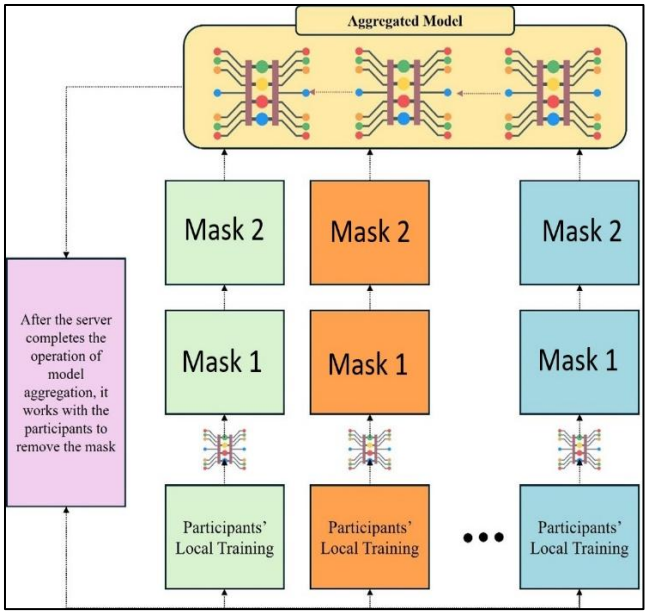


Figure 2: Basic double masking in FL.

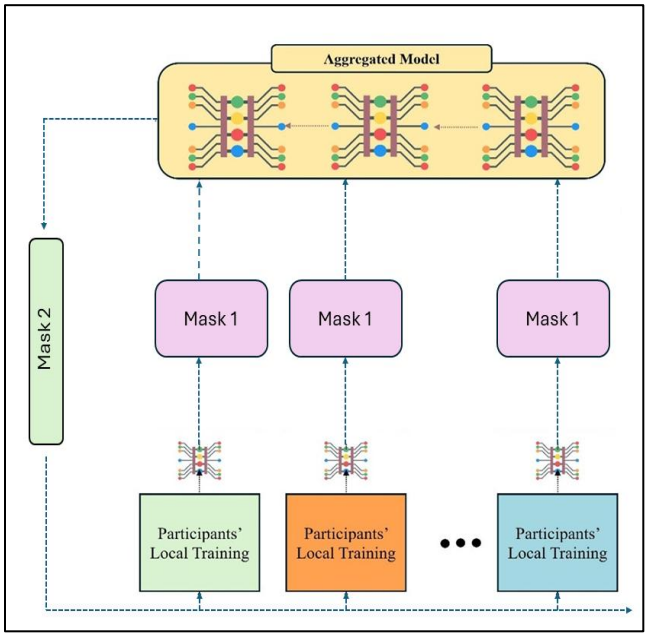


Figure 3: Double masking as a Dual-Lock Security in FL.

3.1.PRNG Client-Side Mask

The concept of Pseudo-Random Number Generator (PRNG) is explored through three main lenses: mathematical enhancement of chaos, hardware-specific optimization (FPGA/RFID), and practical privacy applications (Smart Grids). In the context of the smart grid ecosystem, the PRNG serves as the foundation for a "privacy shield" through a technique known as data masking. In this framework, both the individual smart meter and the utility company share a synchronized PRNG key. The smart meter generates a continuous stream of pseudorandom values, or "masks," which are added to the consumer's actual energy usage data before transmission. To an outside eavesdropper or unauthorized third party, the resulting data appears as meaningless noise, effectively hiding granular usage patterns that could reveal private information, such as when a resident is home or which specific appliances are in use. Because the utility provider possesses the identical PRNG key, it can generate the same sequence of random numbers and subtract them from the received signal to recover the original, accurate energy totals for billing and grid management.

PRNG has been utilized as a data mask, and it has been defined as a deterministic cryptographic primitive that transforms sensitive plaintext into a high-entropy sequence through additive obfuscation. This process functions similarly to a stream cipher, where the PRNG leverages a secret seed shared exclusively between the sender and the intended recipient to produce a computationally indistinguishable sequence of random values. By combining these values with the original data, typically via modular addition or bitwise XOR operations, the system creates a "mask" that decouples the user's behavioral patterns from the transmitted signal. The security of this mechanism relies on the mathematical inability of an adversary to distinguish the masked data from true noise without possession of the seed, ensuring that the recipient can accurately reconstruct the original information by applying an inverse operation while maintaining total data confidentiality during transit [56-60]. In the MASKER protocol and similar privacy-preserving frameworks, the masking process is mathematically defined as a transformation of a sensitive measurement m_i into a masked value x_i using a pseudorandom number r_i. This is expressed by Eq. 4:

$$\boldsymbol{x_i = m_i + r_i \ \ (mod\ N)} \qquad (4)$$

r_i is the output of the PRNG at time i, and N represents the range of the data or the modulus of the cryptographic system. The academic definition characterizes this as an additive homomorphic masking technique. Because the PRNG is deterministic, the utility provider possessing the same seed and algorithm generates the identical value r_i . By performing the inverse operation 5, the provider can extract the exact energy consumption while the raw, unmasked data m_i remains mathematically shielded from all other entities in the network [61].

$$\boldsymbol{x_i = m_i + r_i \ \ (mod\ N)} \qquad (5)$$

3.2.Hash Based Mask Sever-Side Mask

Hash-based mask functions in FL act as a server-side mechanism to enhance the privacy of client updates during secure aggregation. Unlike client-generated masks, this mask is deterministically derived from a cryptographic hash function, often seeded with a shared secret and the training round number, allowing the server to obfuscate aggregated model updates without learning individual contributions. When combined with client-side masking such as additive or double masking the hash-based server mask provides an additional layer of security, ensuring that even if the server is curious or partially compromised, the confidentiality of each client's local update is preserved. This approach is computationally efficient, integrates seamlessly with hybrid privacy-preserving techniques, and contributes to robust, privacy-aware aggregation in FL systems. [62-72].

To protect local model updates from server-side inference and adversarial reconstruction attacks in federated learning, this study adopts a hash-based masking mechanism that integrates cryptographic hash functions with pseudo-random perturbations. Hash-based transformations have been extensively studied for semantic preservation and privacy enhancement in high-dimensional representations [61][62], as well as for secure signal encryption through chaotic and non-linear mappings [63–67]. Inspired by these works, the proposed masking strategy leverages hash-generated randomness to conceal sensitive gradients while maintaining global aggregation correctness.

Let client i compute its local model update at communication round t as:

$$\boldsymbol{\Delta W_i^{(t)} = W_i^{(t)} - W^{(t-1)}} \qquad (6)$$

where $W^{(t-1)}$ denotes the global model distributed by the server. To mask this update, the client generates a pseudo-random mask using a cryptographic hash function:

$$\boldsymbol{M_i^{(t)} = \mathbf{H}(\mathbf{ID}_i \parallel \mathbf{t} \parallel \mathbf{k}_i)} \qquad (7)$$

where $H(\cdot)$ is a secure hash function, ID_i represents the client identity, t is the round index, and k_i is a secret seed locally stored by the client. Hash-based randomness has been shown to preserve semantic consistency while preventing inversion and similarity leakage [61, 62]. The masked update transmitted to the server is then computed as:

$$\boldsymbol{\widetilde{\Delta W}_i^t = \Delta W_i^{(t)} + \lambda . \Phi\left(M_i^{(t)}\right)} \qquad (8)$$

where λ controls the masking strength and $\Phi(\cdot)$ maps the hash output into a real-valued vector compatible with the model dimension. Similar non-linear and chaotic transformations have been proven effective in resisting statistical and differential attacks in secure communication systems [63–67]. At the aggregation stage, the server computes

$$\boldsymbol{W^{(t)} = W^{(t-1)} + \frac{1}{N}\sum_{i=1}^{N} \widetilde{\Delta W}_i^{(t)}} \qquad (9)$$

Due to the independence and zero-mean property of the hash-derived masks, the aggregated noise statistically cancels out when averaged over enough participating clients, ensuring convergence of

the global model. This design principle aligns with recent privacy-preserving FL schemes that balance confidentiality and utility [69–72]. Hash-based masking introduces minimal computational overhead compared to homomorphic encryption or differential privacy mechanisms and still providing strong resistance against gradient leakage and model inversion attacks. combining with sparsification strategies like Top-Q update selection [68] can reduce communication costs without compromising privacy. The proposed hash-based masking framework benefits from the proven security of hash functions [61][62], the robustness of chaotic masking principles [63–67], and the scalability requirements of modern FL systems [69–72] and making it well suited for large-scale, privacy-sensitive distributed learning environments.

3.3.Top-Random Quantization

TRQ is a client selection strategy in FL designed to balance model performance, communication efficiency, and privacy preservation. Unlike deterministic Top-Q approaches, which select clients with the largest gradient norms or contributions, TRQ introduces stochasticity by randomly selecting a subset from the top-ranked clients. This mitigates selection bias and reduces privacy leakage from repeated participation.

Formally, let $C = \{1,2, \dots, N\}$denote the set of all clients, and let ΔW_i^tbe the local model update from client $i$at round t. In Top-Q selection, the server chooses the top Qclients based on the norm of updates:

$$\| \boldsymbol{\Delta W_i^t} \|_2, \boldsymbol{i} \in \boldsymbol{C} \qquad (\mathbf{10})$$

For Top-Random-Q, instead of deterministically choosing the top Qclients, we define the top-K candidate set C_{top}^t, where $K \geq Q$. Then, a random subset of size $Q$is selected as:

$$\boldsymbol{S^t} = \text{RandomSubset}\left(\boldsymbol{C_{\text{top}}^t}, \boldsymbol{Q}\right) \qquad (\mathbf{11})$$

The global model is updated using only the selected clients:

$$\boldsymbol{W^{t+1}} = \boldsymbol{W^t} + \frac{\mathbf{1}}{\boldsymbol{Q}} \sum_{\boldsymbol{i} \in \boldsymbol{S^t}} \boldsymbol{\Delta W_i^t} \qquad (\mathbf{12})$$

This stochastic selection improves privacy, as repeated exposure of the same client updates is minimized, while maintaining model accuracy through inclusion of high-importance contributions. Recent studies demonstrate the advantages of TRQ in privacy-preserving FL. DPSHE integrates homomorphic encryption and differential privacy for medical imaging, showing that randomized selection reduces inference attacks on private data [68]. Similarly, privacy-preserving FL for IoT intrusion detection benefits from stochastic selection to enhance real-time security across devices [69]. Its use in decentralized NIR spectroscopy calibration preserves accuracy while maintaining privacy [70]. Furthermore, combining quantum computing with FL indicates that random selection contributes to security resilience and privacy preservation in IoT networks [71].

3.4.Adaptive Moment Estimation (ADAM)

Adam is one of the most widely used optimization algorithms in ML and DL. ADAM can overcome the limitations of traditional stochastic gradient descent (SGD), especially when gradients are sparse, noisy, or highly variable [72]. Combining momentum-based updates with adaptive learning rates, Adam allows for faster convergence and more stable training, making it particularly effective for large-scale and deep neural network models [73][74].

At each iteration t, let $\theta_t$represent the model parameters, and let represent the mathematical equation as follows:

$$\boldsymbol{g_t = \nabla_\theta \mathcal{L}(\theta_t)} \qquad \textbf{(13)}$$

Where ∇_θ denotes the gradient of the loss function $\mathcal{L}$ with respect to these parameters. ADAM maintains exponentially weighted moving averages of both the first moment (mean) and the second moment (uncentered variance) of the gradients. These are computed as [72]:

$$\boldsymbol{m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t} \qquad \textbf{(14)}$$
$$\boldsymbol{v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2} \qquad \textbf{(15)}$$

Both $\boldsymbol{\beta_1}$and $\boldsymbol{\beta_2}$are the decay rates controlling the contribution of historical gradients, typically set to $\boldsymbol{0.9}$and $\boldsymbol{0.999}$, respectively. The first moment $\boldsymbol{m_t}$captures the general direction of the gradient, while the second moment $\boldsymbol{v_t}$captures its magnitude [73].

Since the moment estimates are initialized at zero, they are biased toward zero during the initial training steps. To correct this bias, ADAM applies the following bias-correction terms [72]:

$$\boldsymbol{\hat{m}_t = \frac{m_t}{1-\beta_1^t}, \hat{v}_t = \frac{v_t}{1-\beta_2^t}} \qquad \textbf{(16)}$$

The model parameters are then updated according to:

$$\boldsymbol{\theta_{t+1} = \theta_t - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t}+\varepsilon}} \qquad \textbf{(17)}$$

where $\boldsymbol{\alpha}$is the base learning rate and $\boldsymbol{\varepsilon}$is a small constant added for numerical stability. This adaptive mechanism allows each parameter to have its own effective learning rate, improving optimization in complex, high-dimensional parameter spaces [74–76].

ADAM offers several advantages that have contributed to its widespread adoption. By automatically adjusting the learning rate for each parameter based on both the magnitude and direction of past gradients, ADAM effectively handles sparse, noisy, or highly variable gradient information, which often poses challenges for standard SGD [72][73]. Its use of momentum

through the first-order moment estimate accelerates convergence and reduces oscillations, leading to more stable training, particularly in high-dimensional parameter spaces [74]. In addition, ADAM exhibits robustness to hyperparameter settings, performing well under commonly used defaults, which simplifies experimentation and practical deployment. The optimizer is also computationally efficient, requiring only modest additional memory to maintain moment estimates, making it suitable for large-scale neural networks and datasets [75]. The adaptive and decentralized nature of ADAM renders it particularly effective in distributed and FL contexts, where data heterogeneity across clients can significantly impact the performance of non-adaptive optimization methods [76]. These properties make ADAM a reliable and versatile choice for a wide range of ML applications.

4. Methodology and Methodes

This research employs an experimental quantitative approach to evaluating FL privacy preserving. The methodology integrates local model training with a cryptographic masking protocol to protect client privacy during server aggregation. Research Design and Architecture The study utilizes a decentralized training architecture where N clients N=5 collaboratively train a global model without sharing raw data. The architecture consists of two primary components Global Server that Responsible for model distribution and the aggregation of masked updates and Local Clients Responsible for local training on private data and applying pairwise masks to the resulting model updates.

4.1.Dataset and Preprocessing

In this study, experimental results were conducted using the MNIST handwritten digit dataset, a widely adopted benchmark for image classification tasks. The dataset consists of 70,000 grayscale images representing ten-digit classes (0–9) and provides a controlled and reproducible environment for assessing model performance under FL settings.

To approximate a realistic yet stable FL scenario, the dataset was distributed across five participating clients following an IID sampling strategy. The MNIST dataset was randomly shuffled to mitigate any potential ordering bias. It was then partitioned into five equal subsets, ensuring that each client received a statistically representative portion of the global data distribution. This balanced allocation supports stable convergence behavior and facilitates fair performance evaluation during decentralized training. A uniform preprocessing pipeline was applied consistently across all clients. Each image was transformed into a tensor format and normalized using a global mean value of 0.1307 and a standard deviation of 0.3081. Such normalization is essential for harmonizing feature scales across distributed clients and enhancing numerical stability during optimization. By centering and scaling the input data, the training process becomes less sensitive to gradient fluctuations, thereby improving the reliability of local model updates and enhancing the effectiveness of global aggregation across successive communication rounds.

4.2.Proposed Framework based on FL and Double Masking Technique

The classification component of the proposed framework is structured CNN architecture with the objective of achieving an optimal balance between privacy preserving and high accuracy. Network

architecture started with a sequential feature extraction module content of two convolutional layers, each employing 3×3 kernels to effectively capture local spatial patterns present in handwritten digit images. These convolutional layers are followed by Rectified Linear Unit (ReLU) activation functions to introduce non-linearity, as well as 2×2 max-pooling operations to progressively reduce the spatial resolution of the feature maps. This design choice not only lowers computational complexity but also enhances robustness to minor spatial variations in the input data.

Following the convolutional stages, the extracted feature maps are flattened into a one-dimensional vector and passed to a fully connected hidden layer consisting of 128 neurons. This dense layer functions as the primary decision-making component, integrating the high-level features learned during the convolutional process. Finally, the network outputs are produced through a softmax-activated layer with ten neurons, each representing a distinct digit class from 0 to 9. The softmax function generates a normalized probability distribution over the output classes, enabling the model to assign each input image to the class with the highest predicted probability.

To safeguard individual client updates against potential server-side inspection, this study employed a privacy-preserving protocol centered on a PRNG masking scheme. This mechanism is founded on the establishment of shared cryptographic seeds between adjacent participants. For each communication round t, every client i generates a common seed with its immediate neighbor $i+1$ via a simulated hash-based agreement. By utilizing the SHA-256 algorithm from the hashlib library, a unique, reproducible identifier is derived for each client pair, ensuring that the generated noise remains consistent between the two parties without requiring a persistent security channel for every update. When seeds are established, clients generate stochastic noise tensors, denoted as M which are precisely tailored to match the dimensionality of the model's parameter space. To maintain the utility of the global model, masks are calibrated using a scale factor of 1×10^{-4}providing a judicious compromise between gradient obscuration and numerical stability. Masking operation follows a cyclic additive-subtractive logic where each client i computes a perturbed update U'_i by incorporating a forward mask shared with client i+1 and removing a backward mask shared with client i-1. This process is formally defined by Eq.15:

$$\boldsymbol{U'_i = U_i + M_{i,i+1} - M_{i-1,i}} \qquad (15)$$

The structural design is realized during the server-side aggregation phase shown in Eq. 7. While each individual update U'_iremains substantially augmented and mathematically indecipherable to the server, the cumulative sum of the masks across the entire ring of N clients specifically

$$\sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} (\boldsymbol{M_{i,i+1} - M_{i-1,i}}) \text{ yields zero.} \qquad (\mathbf{16})$$

The server can compute the true average of the global update without ever gaining access to the raw, sensitive gradients of any single participant. This ensures that the framework achieves the

technical merit of secure aggregation while preserving the operational proficiency of the standard FedAvg algorithm.

4.2.1. Training and Aggregation Model

Local training process was performed by using ADAM optimizer, it was selected because of its adaptive moment estimation capability and its effectiveness in promoting stable convergence under distributed training conditions. The learning rate was fixed at η=0.001, a value determined empirically through preliminary experiments to achieve an appropriate balance between convergence speed and optimization stability. To allow each client to adequately refine its local model parameters prior to global synchronization, each participant conducted 20 local training epochs during every communication round. The local model training procedure was executed over 10 global communication rounds, providing sufficient training depth for the global model to capture the underlying characteristics of the MNIST dataset.

At the end of each round, a centralized aggregation phase was sent to the server to integrate the locally learned updates into a unified global model. The Federated Averaging (FedAvg) algorithm was employed for this purpose, wherein the server computes a weighted average of the received masked updates U_i'.Each client's contribution was weighted according to the number of local training samples n_i, ensuring that clients with larger datasets exert a proportionally greater influence on the global update. The aggregation process is defined as shown in Eq.8:

$$\boldsymbol{W_{t+1} = W_{t+1} + \sum_{i=1}^{N} \frac{n_i}{n_{total}} U_i'} \qquad (\mathbf{17})$$

Where W_t and W_{t+1} denote the global model parameters at communication rounds t and $t+1$, respectively. By combining the proposed pairwise masking protocol with this weighted aggregation rule, the server effectively eliminates the masking noise during aggregation while preserving the correctness of the global update. Consequently, the resulting global model parameters are equivalent to those obtained in a non-privacy-preserving setting, thereby maintaining model utility without exposing individual client updates or local data distributions.

4.2.2. Evaluation Metrics

The effectiveness of the proposed framework is measured by quantifying the system with four distinct analytical lenses, ensuring a comprehensive understanding of how the privacy mechanism impacts the overall ML pipeline as shown in Table 3. The first and most critical metric is the model's predictive performance, quantified by monitoring both the test accuracy and the cross-

entropy loss on a global validation set. This allows for a clear observation of whether the added privacy layers interfere with the model's ability to converge on an accurate solution. To empirically measure the strength of the privacy-preserving layer, the model tracked the Privacy Leakage by calculating the L2 norm of the update vectors at two critical stages: immediately following local training and again after the pairwise masks were applied. Markedly Heightened L2 norm for the masked updates serves as a proxy for successful privacy preservation, it demonstrates that the sensitive, raw gradients have been effectively obscured by stochastic noise before leaving the client's local environment. Beyond model performance and security considered the practical constraints of a distributed system by measuring communication overhead and computational complexity. Communication traffic was calculated as the total volume of data in megabytes transmitted between the clients and the central server to identify any bandwidth bottlenecks introduced by the protocol, and the computational time required for both the local training epochs and the generation of the cryptographic masks was recorded. Analyzing the four metrics synergistically, it was allowed to identify the optimal balance between the rigorous demands of data privacy and the operational requirements of a functional FL system.

Table 3: Evaluation Metrics

Metric	Definition / Purpose	Measurement Method
Test Accuracy	Measures the predictive performance of the global model	Evaluate the model on the test set after each round
Computational Overhead	Time taken for training and mask generation per round	Record total time per round
Communication Traffic	Volume of data sent between clients and server per round	Calculate total parameters × 4 bytes × number of clients, converted to MB
Privacy Leakage	Privacy Raw L2 Norm Magnitude of raw updates before masking	Compute L2 norm of concatenated local updates for each client
	Privacy Masked L2 Norm Magnitude of updates after TQR + client masking	Compute L2 norm of masked updates
	Calculate privacy improvement from L2 norm and masked L2	

4.2.3. Proposed Model

This study proposes a privacy-preserving FL framework integrates FedAvg aggregation, ADAM optimizer, TRQ sparsification, and a double masking mechanism combining PRNG-based client-side masking with hash-based server-side masking. The primary objective of the proposed model is to prevent the central server from inferring sensitive information from individual client updates, while maintaining model utility and convergence performance. The system follows a classical cross-device FL architecture, consisting of a central server and multiple distributed clients. Each client trains a local model on its private dataset and transmits only masked model updates to the

server. The server receives clients' updates, Applies Hash-Based Server Mask and aggregates these masked updates to obtain a global model without accessing any individual client's raw update.

I. Local Model training started as follows in Eqs. [9-22]. Each client $i$trains a local model $\mathrm{w}_i$initialized from the global model $\mathrm{w}_t$at round t. Unlike standard SGD, we employ the ADAM optimizer to allow faster convergence and better stability, and it adaptively adjusts learning rates for each parameter using first and second moment estimates of gradients:

$$\boldsymbol{m_i^t} = \boldsymbol{\beta_1 m_i^{t-1}} + (\mathbf{1}-\boldsymbol{\beta_1})\boldsymbol{\nabla \mathcal{L}_i(\mathbf{w}_i^t)} \quad (\mathbf{18})$$

$$\boldsymbol{v_i^t} = \boldsymbol{\beta_2 v_i^{t-1}} + (\mathbf{1}-\boldsymbol{\beta_2})(\boldsymbol{\nabla \mathcal{L}_i(\mathbf{w}_i^t)})^{\mathbf{2}} \quad (\mathbf{19})$$

$$\boldsymbol{\widehat{m}_i^t} = \frac{\boldsymbol{m_i^t}}{\mathbf{1}-\boldsymbol{\beta_1^t}}, \boldsymbol{\widehat{v}_i^t} = \frac{\boldsymbol{v_i^t}}{\mathbf{1}-\boldsymbol{\beta_2^t}} \quad (\mathbf{20})$$

$$\mathbf{w_i^{t+1}} = \mathbf{w_i^t} - \boldsymbol{\eta}\frac{\boldsymbol{\widehat{m}_i^t}}{\sqrt{\boldsymbol{\widehat{v}_i^t}}+\boldsymbol{\epsilon}} \quad (\mathbf{21})$$

Where $\eta$is the learning rate,$\beta_1, \beta_2$are the exponential decay rates for the moment estimates, ϵ is a small constant to prevent division by zero,$\mathcal{L}_i$is the local cross-entropy loss for client i.The local update sent to the server is:

$$\Delta \mathrm{w}_i^t = \mathrm{w}_i^{t+1} - \mathrm{w}_t \quad (\mathbf{22})$$

II. Applying TQRR instead of selecting a purely random subset of parameters, TQRR first identifies the Top-Q fraction of parameters by absolute magnitude, then randomly retains a fraction R of these Top-Q parameters. TQRR preserves the most significant updates while introducing controlled randomness to reduce privacy leakage and maintain communication efficiency. Mathematically, for client i at round t:

$$\Delta \mathrm{w}_i^{t,\mathrm{TQRR}} = \Delta \mathrm{w}_i^t \odot \mathrm{M}_i^{t,\mathrm{TQRR}} \quad \mathbf{(23)}$$

Where $\mathrm{M}_i^{t,\mathrm{TQRR}}$is the mask selecting R fraction of Top-Q elements.

III. To apply privacy preserving, Client-Side PRNG Masking to enhance privacy, each client generates a pseudo-random mask using a deterministic PRNG seeded by the client ID and round number:

$$\mathrm{P}_i^t = \mathrm{PRNG}(i,t) \in \mathbb{R}^N, \mathrm{P}_i^t \sim \mathrm{Uniform}(-\alpha, \alpha) \quad \mathbf{(24)}$$

Where $\alpha$is the mask scale. The masked update is:

$$\Delta \mathrm{w}_i^{t,\mathrm{masked}} = \Delta \mathrm{w}_i^{t,\mathrm{sparse}} + \mathrm{P}_i^t \quad \mathbf{(25)}$$

This ensures that the server cannot reconstruct raw client updates.

IV. Server-Side Hash-Based Masking is applied after aggregating client updates using FedAvg, the server temporarily applies a hash-based mask for security:

$$\mathrm{H}^{t} = \mathrm{HashMask}(t), \mathrm{H}^{t} \sim \mathrm{Uniform}(-\beta, \beta) \quad (\mathbf{26})$$

$$\Delta \mathrm{w}^{t,\text{server-masked}} = \frac{1}{\sum_i n_i} \sum_{i=1}^{K} n_i \, \Delta \mathrm{w}_i^{t,\text{masked}} + \mathrm{H}^{t} \quad (\mathbf{27})$$

Where $n_i$is the number of local samples for client i, and $K$is the number of clients. The server mask is removed before updating the global model:

$$\mathrm{w}_{t+1} = \mathrm{w}_t + \left(\Delta \mathrm{w}^{t,\text{server-masked}} - \mathrm{H}^{t}\right) \quad \mathbf{(28)}$$

V. Privacy leakage measurement using the L2 norm of updates will apply before and after masking:

$$\mathrm{PrivacyRaw}_i^t = \| \Delta \mathrm{w}_i^t \|_2, \mathrm{PrivacyMasked}_i^t = \| \Delta \mathrm{w}_i^{t,\text{masked}} \|_2 \quad \mathbf{(29)}$$

A lower norm after masking indicates better privacy preservation.

VI. Calculating the communication traffic using communication and computation metrics per round:

$$\mathrm{Traffic}^t = \frac{\sum_{i=1}^{K} |\Delta \mathrm{w}_i^{t,\text{sparse}}| \cdot 32}{8 \cdot 10^6} \text{ MB (assuming 32-bit floats).} \quad \mathbf{(30)}$$

Computational overhead per round can be measured as elapsed time to compute local updates and aggregation:

$$\mathrm{CompOverhead}^t = T_{\text{local}} + T_{\text{aggregation}} \quad \mathbf{(31)}$$

As shown in Algorithm 1, which presents the proposed FL framework incorporating a dual privacy masking strategy. The algorithm begins with client-side local model training using the Adam optimizer, followed by the application of TQRR sparsification to reduce update dimensionality and communication overhead. To enhance privacy at the client level, PRNG based masking is applied to the sparse model updates before transmission. On the server side, masked client updates are aggregated using FedAvg scheme, after which an additional hash-based masking mechanism is employed to further protection of the aggregated gradients. The algorithm evaluates performance metrics, including model accuracy, computational overhead, communication cost, and privacy leakage, across successive communication rounds.

Algorithm 1: Proposed Framework based on FL and Double Masking Technique

for round = 1 to ROUNDS do

Initialize lists: MaskedUpdates, SampleCounts, RawNorms, MaskedNorms

```
    for each client i = 1 to NUM_CLIENTS do
        (LocalModel, NumSamples) ← TrainLocal(GlobalModel,
ClientLoader[i])
        SampleCounts.append(NumSamples)

        RawUpdate ← LocalModel − GlobalModel
        RawNorms.append(||RawUpdate||₂)

        SparseUpdate ← TopRandomQ(RawUpdate, TOPQ_RATIO)

        ClientMask ← PRNGClientMask(i, round, SparseUpdate)
        MaskedUpdate ← SparseUpdate + ClientMask
        MaskedNorms.append(||MaskedUpdate||₂)

        MaskedUpdates.append(MaskedUpdate)
    end for

    AggregatedUpdate ← FedAvg(MaskedUpdates, SampleCounts)

    ServerMask ← HashServerMask(round, AggregatedUpdate)
    ServerMaskedUpdate ← AggregatedUpdate + ServerMask

    FinalUpdate ← ServerMaskedUpdate − ServerMask
    GlobalModel ← GlobalModel + FinalUpdate

    Accuracy ← Evaluate(GlobalModel)
    CommunicationCost ← ComputeTraffic(AggregatedUpdate)

    Record metrics: Accuracy, computation time, communication cost,
RawNorms, MaskedNorms
end for
```

5. Results and Discussion

This section presents the experimental results obtained from the proposed framework based on FL and double masking technique over ten global communication rounds. The evaluation focuses on four main aspects: accuracy, computational overhead, communication traffic, and privacy leakage. All results are reported based on the same experimental setup and training configuration.

5.1.Accuracy Metric

As shown in Table 4, in the first communication round, the model reached an accuracy of 71.84%, it reflects the effect of training on limited local updates. A Significant improvement is observed in

the second round, the accuracy increased to 94.77%, it showed that the aggregation process effectively received useful information from distributed clients. The proposed model reached a final accuracy of 99.13% by the tenth communication round. Accuracy increased sharply during the early training rounds and began to stabilize after the third round, with only minor gains observed thereafter. This convergence behavior demonstrated that the combination of the hash-based privacy masking mechanism does not adversely affect the model's long-term learning stability. This level of performance is achieved despite the relatively intensive local training configuration of 20 epochs per round, which in decentralized learning environments can often lead to local overfitting or training divergence. The test accuracy is shown in Figure 3.

Table 4: Test Accuracy per 10 rounds

Round	1	2	3	4	5	6	7	8	9	10
Accuracy (%)	71.84	94.77	97.92	98.44	98.61	98.80	98.93	98.98	99.02	99.13

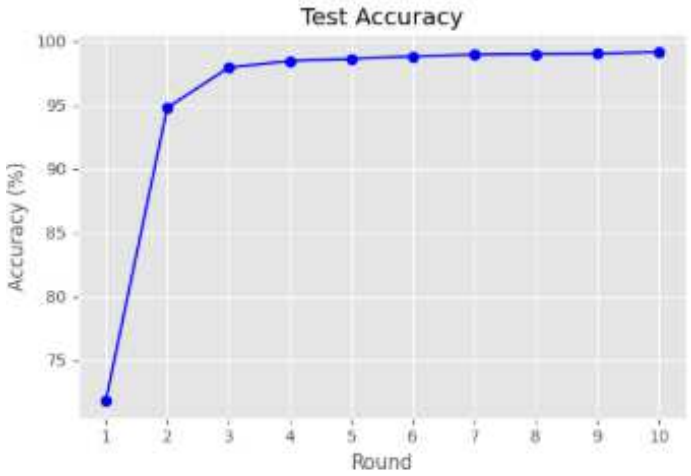


Figure 3: Test accuracy

5.2.Communication Overhead

The experiment results showed an improvement in computational overhead during the training progress. The round execution time decreases from 1017.21 seconds in Round 1 to 929.01 seconds in Round 10 as shown in Table 4, corresponding to an overall overhead reduction of approximately 8.67%.

$$\text{Overhead Improvement (\%)} = \frac{1017.21 - 929.01}{1017.21} \times 100 \approx 8.67\%$$

This improvement in overhead showed that the system benefits from optimizer state stabilization and efficient reuse of computational resources across rounds. Overhead results are shown in Figure 4 and Table 5.

Table 4: Communication Overhead.

Round	Round Time (s)	Overhead (%)
1	1017.21	–
2	935.73	8.03
3	935.79	8.02
4	941.79	7.39
5	943.92	7.23
6	934.19	8.17
7	939.03	7.69
8	930.18	8.56
9	927.75	8.80
10	929.01	8.67

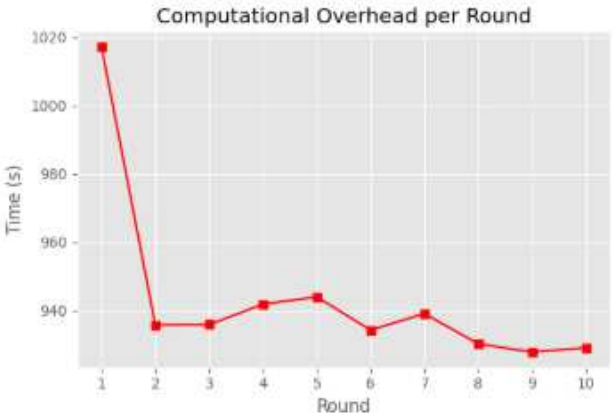


Figure 4: Computational Overhead per Round.

5.3.Communication Traffic

Communication traffic is another evaluation metric in this study. Obviously, the value of communication traffic is stable with a fixed value of 8.04 MB per round across all rounds. It indicates that the proposed FL framework maintains a predictable and controlled communication load throughout the training process. From a practical perspective stable communication traffic simplifies system planning and deployment, especially in bandwidth-limited or large-scale distributed environments. Moreover, achieving high model accuracy while keeping the communication cost unchanged demonstrates that the applied update selection and masking mechanisms are efficient and do not introduce redundant data transmission see Figure 5.

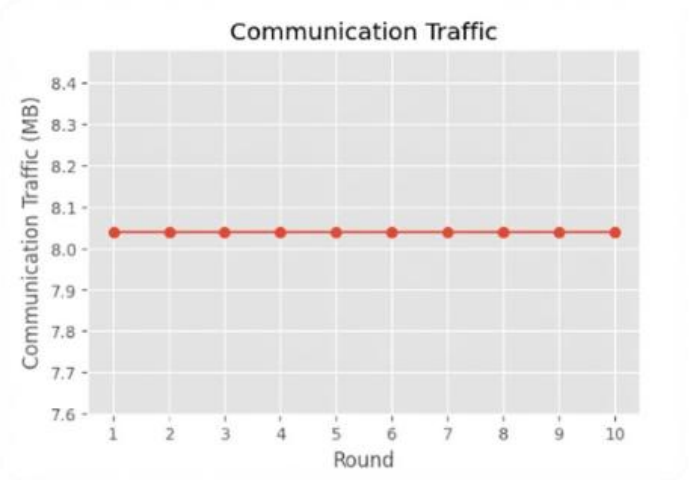


Figure 5: Communication Traffic.

5.4.Privacy leakage

Privacy leakage in this study is evaluated using the L2 norm to measure the amount of sensitive information that could be inferred from model updates before and after applying the masking mechanism. As shown in Figure 6, the average raw L2 norm was 20.86 before applying any masks and decreased to 6.60 after applying the proposed protection mechanisms of this study. This represented an improvement in privacy preserving by approximately 68.3%, highlighting the effectiveness of the masking strategy. The significance of results is reflected in their ability to Substantiate that combining Top-Random-Q selection, PRNG-based randomness, and hash-based masking effectively reduces the amount of information that can be leaked without negatively impacting model accuracy or stability. This substantial decrease in privacy leakage confirms that the proposed framework provided strong protection against gradient inference and data reconstruction attacks and maintaining learning efficiency, making it suitable for real-world privacy-sensitive FL applications.

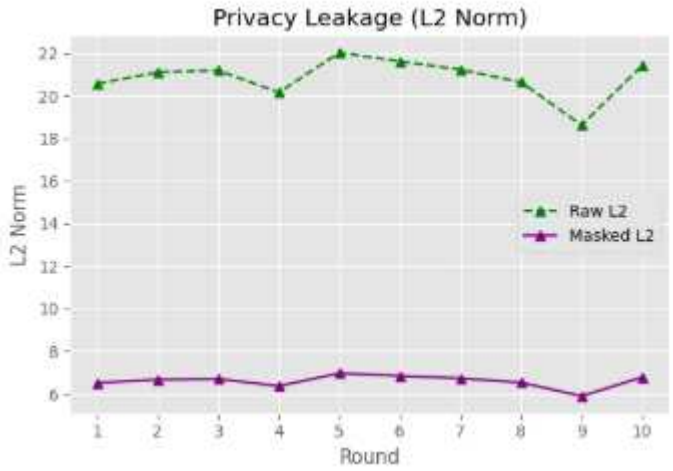


Figure 6: Privacy Leakage.

6. Conclusion and Future Works

This paper introduced a robust and privacy-preserving FL framework that integrates a double masking strategy as client-side PRNG masking combined with server-side hash-based masking, and the ADAM optimizer for efficient local model training. proposed model effectively addressed the critical challenges of federated learning, including privacy leakage, communication efficiency, and training stability, while preserving high model accuracy. A central contribution of this work is the adoption of a double masking mechanism, which provides dual-layer protection against information leakage. Client-side PRNG mask ensures that local updates are obfuscated before transmission and server-side hash-based mask secures the aggregation process. The effectiveness of this strategy in privacy leakage is calculated by the low L2 norm of the masked updates to be improved by 68%, highlighting its robustness against gradient inference and reconstruction

attacks. Using TRQ sparsification further enhances communication efficiency by transmitting only the most informative gradients while maintaining stable and predictable communication traffic across all federated rounds. ADAM optimizer contributes to faster convergence and improved stability under heterogeneous data distributions, enabling the global model to reach a high classification performance. Experimental results on MNIST dataset demonstrate that the proposed model achieved a high accuracy of 99.13%, with reducing in computational overhead of approximately 7.25%. Superiority of the proposed method becomes evident where accuracy of 96.46% was achieved with a high privacy coefficient of 8 by reducing the privacy coefficient to 2 led to a noticeable accuracy degradation to 94.00%, revealing a strong trade-off between privacy and performance. In contrast, the proposed framework maintains both higher accuracy and stronger privacy guarantees simultaneously, without relying on excessive noise injection. This demonstrates that the proposed method successfully mitigates the privacy accuracy trade-off that commonly affects privacy-preserving FL systems.

In the future, there are some points may be added to this framework to achieve more success in privacy preserving and improving overhead and communication traffic.

First, training the framework on different datasets such as CIFAR-10 and CIFAR-100 or real-world datasets for hospitals or universities to ensure the strength of the model.

Second, using different sparsification methods and comparing with them to find the best one in improving the overhead and communication traffic.

Finally, double masking strategy effectively reduces privacy leakage from gradient inference and reconstruction attacks, a deeper security analysis against stronger adversarial models is needed. including scenarios involving colluding servers, Byzantine or malicious clients, and model poisoning attacks, which are common concerns in real-world FL systems.

Data Availability Statement

The MNIST dataset used in this study are publicly available. It can be accessed at https://www.kaggle.com/datasets/hojjatk/mnist-dataset.

Funding Declaration

Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB).

References

1. Gheisari, M., et al. (2023). Deep learning: Applications, architectures, models, tools, and frameworks—A comprehensive survey. *CAAI Transactions on Intelligence Technology, 8*(3), 581–606. https://doi.org/10.1049/cit2.12180
2. Jabeen, S., Li, X., Amin, M. S., Bourahla, O., Li, S., & Jabbar, A. (2023). A review on methods and applications in multimodal deep learning. *ACM Transactions on Multimedia Computing, Communications, and Applications, 19*(2s), 1–41. https://doi.org/10.1145/3545572
3. Rigaki, M., & Garcia, S. (2024). A survey of privacy attacks in machine learning. *ACM Computing Surveys, 56*(4), 1–34. https://doi.org/10.1145/3624010
4. Zhou, J., et al. (2024). PPML-Omics: A privacy-preserving federated machine learning method protects patients' privacy in omic data. *Science Advances, 10*(5), eadh8601. https://doi.org/10.1126/sciadv.adh8601
5. El Mestari, S. Z., Lenzini, G., & Demirci, H. (2024). Preserving data privacy in machine learning systems. *Computer Security, 137*(2), 103605. https://doi.org/10.1016/j.cose.2023.103605
6. Shawkat, M., Ali, Z.H., Salem, M. *et al.* A robust and personalized privacy-preserving approach for adaptive clustered federated distillation. *Sci Rep* 15, 14069 (2025). https://doi.org/10.1038/s41598-025-96468-8
7. Zhang, C., Xie, Y., Bai, H., Yu, B., Li, W., & Gao, Y. (2021). A survey on federated learning. *Knowledge-Based Systems, 216*, 106775. https://doi.org/10.1016/j.knosys.2021.106775
8. Konečnỳ, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., & Bacon, D. (2016). Federated learning: Strategies for improving communication efficiency. *arXiv Preprint*, arXiv:1610.05492.
9. McMahan, B., Moore, E., Ramage, D., Hampson, S., & Arcas, B. A. Y. (2017). Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the International Conference on Artificial Intelligence and Statistics* (pp. 1273–1282). PMLR.
10. Xu, Z., & Cao, S. (2023). Multi-source data privacy protection method based on homomorphic encryption and blockchain. *Computer Modeling in Engineering & Sciences, 136*(1), 861–881. https://doi.org/10.32604/cmes.2023.025159
11. Yang, Q., et al. (2023). Federated learning with privacy-preserving and model IP-right-protection. *Machine Intelligence Research, 20*(1), 19–37. https://doi.org/10.1007/s11633-022-1343-2
12. Yazdinejad, A., Dehghantanha, A., Karimipour, H., Srivastava, G., & Parizi, R. M. (2024). A robust privacy-preserving federated learning model against model poisoning attacks. *IEEE Transactions on Information Forensics and Security, 19*, 6693–6708. https://doi.org/10.1109/TIFS.2024.3420126
13. Badr, M. M., et al. (2023). Privacy-preserving and communication-efficient energy prediction scheme based on federated learning for smart grids. *IEEE Internet of Things Journal, 10*(9), 7719–7736. https://doi.org/10.1109/JIOT.2022.3230586
14. Zhou, X., et al. (2023). Decentralized P2P federated learning for privacy-preserving and resilient mobile robotic systems. *IEEE Wireless Communications, 30*(2), 82–89. https://doi.org/10.1109/MWC.004.2200381

15. Zhang, J., Xue, N., Li, X., Long, Y., Li, E., Xu, C., ... & Chen, F. (2025, January). Federated learning under differential privacy with effective clipping and analytic Gaussian mechanism. In *Fourth International Conference on Network Communication and Information Security (ICNCIS 2024)* (Vol. 13516, pp. 314-322). https://doi.org/10.1117/12.3052001.
16. Chen, J., Xue, J., Wang, Y., Huang, L., Baker, T., & Zhou, Z. (2023). Privacy-preserving and traceable federated learning for data sharing in industrial IoT applications. *Expert Systems with Applications, 213*, 119036. https://doi.org/10.1016/j.eswa.2022.119036
17. Wang, W., Li, X., Qiu, X., Zhang, X., Brusic, V., & Zhao, J. (2023). A privacy preserving framework for federated learning in smart healthcare systems. *Information Processing & Management, 60*(1), 103167. https://doi.org/10.1016/j.ipm.2022.103167
18. Song, J., Wang, W., Gadekallu, T. R., Cao, J., & Liu, Y. (2022). EPPDA: An efficient privacy-preserving data aggregation federated learning scheme. *IEEE Transactions on Network Science and Engineering, 10*(5), 3047–3057. https://doi.org/10.1109/TNSE.2022.3153519
19. Namakshenas, D., Yazdinejad, A., Dehghantanha, A., & Srivastava, G. (2024). Federated quantum-based privacy-preserving threat detection model for consumer Internet of Things. *IEEE Transactions on Consumer Electronics*, 1. https://doi.org/10.1109/TCE.2024.3377550
20. Ma, J., Naas, S., Sigg, S., & Lyu, X. (2022). Privacy-preserving federated learning based on multi-key homomorphic encryption. *International Journal of Intelligent Systems, 37*(9), 5880–5901. https://doi.org/10.1002/int.22818
21. Zhang, L., Xu, J., Vijayakumar, P., Sharma, P. K., & Ghosh, U. (2022). Homomorphic encryption-based privacy-preserving federated learning in IoT-enabled healthcare system. *IEEE Transactions on Network Science and Engineering, 10*(5), 2864–2880. https://doi.org/10.1109/TNSE.2022.3185327
22. Nguyen, T., & Thai, M. T. (2023). Preserving privacy and security in federated learning. *IEEE/ACM Transactions on Networking, 32*(1), 1–11. https://doi.org/10.1109/TNET.2023.3302016
23. Liu, X., Sun, H., Sun, N., Jia, B., & Xiao, D. (2026). *DPSHE: A privacypreserving federated learning scheme based on homomorphic encryption and differential privacy for medical image. Neurocomputing, 661*, 131931. https://doi.org/10.1016/j.neucom.2025.131931
24. Song, Z., Huang, Z., & Cai, Y. (2026). *A verifiable privacypreserving federated learning scheme based on homomorphic proxy reencryption. Information Sciences, 730*, 122875. https://doi.org/10.1016/j.ins.2025.122875
25. Rafiee, M., & Seyedghasemi, F. (2026). *Privacy attack on a federated learning scheme. Computers and Electrical Engineering, 129*(Part A), 110830. https://doi.org/10.1016/j.compeleceng.2025.110830
26. Din, S., Imran, M., Irfan, S., Tariq, A., Elgamal, Z., & Dawood, H. (2026). *Federated learning for melanoma detection: A decentralized and privacypreserving approach using deep learning. Biomedical Signal Processing and Control, 112*(Part A), 108576. https://doi.org/10.1016/j.bspc.2025.108576
27. Qin, X., Yang, X., & Tang, X. (2026). *Practical privacypreserving federated learning based on multiparty homomorphic encryption for largescale models. Pattern Recognition, 171*(Part B), 112174. https://doi.org/10.1016/j.patcog.2025.112174

28. Islam, M. S. ul, & Wali, A. (2026). *GeoFedNet: Federated learning for privacyaware, robust, and generalizable seismic interpretation. Computers & Geosciences, 207*, 106060. https://doi.org/10.1016/j.cageo.2025.106060
29. Wang, X., Liu, Z., Dai, M., Gong, J., & Ni, W. (2026). *A verifiable and efficient chained federated learning scheme for privacy protection. Computer Networks, 274*, 111838. https://doi.org/10.1016/j.comnet.2025.111838
30. Puviarasu, A., & Sudha, V. K. (2026). *Enhanced IoT security: privacypreserving federated learning model for accurate, realtime intrusion detection across devices. Ain Shams Engineering Journal, 17*(1), 103866. https://doi.org/10.1016/j.asej.2025.103866
31. Chen, Y.-y. (2026). *Decentralized federated learning enables privacypreserving NIR spectroscopy calibration: A proofofconcept study. Chemometrics and Intelligent Laboratory Systems, 268*, 105583. https://doi.org/10.1016/j.chemolab.2025.105583
32. R. Ganesh Babu, Geetha T. S., Nedumaran, A., Sugumaran, B., & Muralikrishnan, P. (2026). *Integrating quantum computing with federated learning for enhanced security and privacy in IoT networks. Results in Engineering, 29*, 108500. https://doi.org/10.1016/j.rineng.2025.108500
33. Liu, P., Zhang, M., Shen, Z., & Wang, H. (2026). *Federated learning with privacy protection and incentive mechanisms based on edge computing. Future Generation Computer Systems, 176*, 108216. https://doi.org/10.1016/j.future.2025.108216
34. Wu, Y., Ren, Y., Guo, Z., & Huang, M. (2026). *Privacypreserving personalized federated prompt learning for visionlanguage models. Neural Networks, 195*, 108220. https://doi.org/10.1016/j.neunet.2025.108220
35. Mai, J., Liu, Z., Guo, J., Miao, Y., Chen, Y., Wang, L., & Ma, J. (2026). *FedHSA: A privacypreserving federated learning framework with heterogeneous system alignment. Information Fusion, 128*, 103920. https://doi.org/10.1016/j.inffus.2025.103920
36. Tran, T.-D., Bui, P.-D., Cam, N. T., & Pham, V.-H. (2026). *Fedvuln: Scalable and privacypreserving federated graph learning for smart contract vulnerability detection on parallel systems. Future Generation Computer Systems, 178*, 108264. https://doi.org/10.1016/j.future.2025.108264
37. Jiang, X., Qian, J., Gu, A., Ma, X., Jin, K., Zhang, X., & Song, Z. (2026). *SynthFed: Privacypreserving longtail ophthalmic diagnosis via VQVAE and GPTaugmented federated learning. Biomedical Signal Processing and Control, 113*(Part D), 109181. https://doi.org/10.1016/j.bspc.2025.109181
38. Ogier du Terrail, J., Ayed, S.-S., Cyffers, E., Grimberg, F., He, C., Loeb, R., Mangold, P., Marchand, T., Marfoq, O., Mushtaq, E., Muzellec, B., Philippenko, C., Silva, S., Teleńczuk, M., Albarqouni, S., Avestimehr, S., Bellet, A., Dieuleveut, A., Jaggi, M., Karimireddy, S. P., Lorenzi, M., Neglia, G., Tommasi, M., & Andreux, M. (2022). FLamby: Datasets and benchmarks for cross-silo federated learning in realistic healthcare settings. *arXiv preprint* arXiv:2210.04620. https://doi.org/10.48550/arXiv.2210.04620
39. Moshawrab, M., Adda, M., Bouzouane, A., Ibrahim, H., & Raad, A. (2023). Reviewing Federated Learning Aggregation Algorithms; Strategies, Contributions, Limitations and Future Perspectives. *Electronics*, *12*(10), 2287. https://doi.org/10.3390/electronics12102287
40. Zhang, H., Zhang, P., Hu, M., Liu, M., & Wang, J. (2024). FedUB: Federated Learning Algorithm Based on Update Bias. *Mathematics*, *12*(10), 1601. https://doi.org/10.3390/math12101601

41. Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., & Suresh, A. T. (2020). SCAFFOLD: Stochastic controlled averaging for federated learning. In H. D. III & A. Singh (Eds.), *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)* (Vol. 119, pp. 5132–5143). PMLR. http://proceedings.mlr.press/v119/karimireddy20a.html

42. Quintana, G. I., Vancamberg, L., Jugnon, V., Mougeot, M., & Desolneux, A. (2024). *BN-SCAFFOLD: Controlling the drift of Batch Normalization statistics in Federated Learning*. arXiv preprint arXiv:2410.03281. https://doi.org/10.48550/arXiv.2410.03281

43. Wang, H., et al. (2020). *Federated learning with matched averaging*. ICLR. https://arxiv.org/abs/2002.06440

44. Ek, S., Portet, F., Lalanda, P., & Vega, G. (2021). A federated learning aggregation algorithm for pervasive computing: Evaluation and comparison. In *2021 IEEE International Conference on Pervasive Computing and Communications (PerCom)* (pp. 1–10). IEEE. https://doi.org/10.1109/PERCOM50583.2021.9439129

45. Wang, H., Yurochkin, M., Sun, Y., Papailiopoulos, D., & Khazaeni, Y. (2020). Federated learning with matched averaging. In *Proceedings of the 8th International Conference on Learning Representations (ICLR 2020)*. https://arxiv.org/abs/2002.06440

46. Reddi, S. J., et al. (2021). *Adaptive federated optimization*. ICLR. https://arxiv.org/abs/2003.00295

47. Theologitis, M., Samoladas, V., & Deligiannakis, A. (2025). *FDA-Opt: Communication-Efficient Federated Fine-Tuning of Language Models via Dynamic Update Schedules* [Preprint]. arXiv. https://arxiv.org/abs/2505.04535

48. Karami, M., Ghassemi, F., Kebriaei, H., & Azadegan, H. (2025). *OptiGradTrust: Byzantine-Robust Federated Learning with Multi-Feature Gradient Analysis and Reinforcement Learning-Based Trust Weighting*. https://doi.org/10.48550/arXiv.2507.23638

49. Chen, H., Xu, T., Wu, X.-j., & Kittler, J. (2025). Hybrid Batch Normalisation: Resolving the dilemma of batch normalisation in federated learning. *arXiv preprint* arXiv:2505.21877. https://doi.org/10.48550/arXiv.2505.21877

50. Mishra, A., & Lone, H. (2025). *RiM: Record, Improve and Maintain Physical Well-being using Federated Learning* [Bachelor's thesis/report, IISER Bhopal]. arXiv:2505.06384. https://doi.org/10.48550/arXiv.2505.06384

51. Cheong, Y.-N., Shin, K.-S., & Shin, S.-Y. (2025). *Adaptive client sampling and feature separation: Towards efficient personalization in non-IID federated learning*. *Journal of Information and Communication Convergence Engineering, 23*(1). https://doi.org/10.56977/jicce.2025.23.1.1

52. Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., & Seth, K. (2017, October). *Practical secure aggregation for privacy-preserving machine learning.* In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17)* (pp. 1175–1191). Association for Computing Machinery. https://doi.org/10.1145/3133956.3133982.

53. Park, S., Lee, J., Harada, K., & Chi, J. (2025). *Masking and Homomorphic EncryptionCombined Secure Aggregation for PrivacyPreserving Federated Learning. Electronics, 14*(1), 177. https://doi.org/10.3390/electronics14010177
54. Shalabi, E., Khedr, W., Rushdy, E., & Salah, A. (2025). *A Comparative Study of PrivacyPreserving Techniques in Federated Learning: A Performance and Security Analysis. Information, 16*(3), 244. https://doi.org/10.3390/info16030244
55. *Robust and privacypreserving federated learning with distributed additive encryption against poisoning attacks. Computer Networks, 245*, 110383. https://doi.org/10.1016/j.comnet.2024.110383
56. Sharobim, B. K., Yacoub, M. H., Sayed, W. S., Radwan, A. G., & Said, L. A. (2023). *Artificial Neural Network Chaotic PRNG and simple encryption on FPGA. Engineering Applications of Artificial Intelligence, 126*, 106888. https://doi.org/10.1016/j.engappai.2023.10688
57. Valle, J., Machicao, J., & Bruno, O. M. (2022). *Chaotical PRNG based on composition of logistic and tent maps using deepzoom. Chaos, Solitons & Fractals, 161*, 112296. https://doi.org/10.1016/j.chaos.2022.112296
58. PerisLopez, P., HernándezCastro, J. C., EstévezTapiador, J. M., & Ribagorda, A. (2009). *LAMED — A PRNG for EPC Class1 Generation2 RFID specification. Computer Standards & Interfaces, 31*(1), 88–97. https://doi.org/10.1016/j.csi.2007.11.013
59. Elmanfaloty, R. A., & AbouBakr, E. (2019). *Random property enhancement of a 1D chaotic PRNG with finite precision implementation. Chaos, Solitons & Fractals, 118*, 134–144. https://doi.org/10.1016/j.chaos.2018.11.019
60. Karopoulos, G., Ntantogian, C., & Xenakis, C. (2018). *MASKER: Masking for privacypreserving aggregation in the smart grid ecosystem. Computers & Security, 73*, 307–325. https://doi.org/10.1016/j.cose.2017.11.008
61. Li, X., Yu, J., Cheng, J., Li, Z., & Bian, C. (2025). *Semantic PreservationBased Hash Code Generation for finegrained image retrieval. Expert Systems with Applications, 271*, 126668. https://doi.org/10.1016/j.eswa.2025.126668
62. Shang, S., Du, X., Wang, X., & Liu, A. (2025). *Private approximate nearest neighbor search for onchain data based on localitysensitive hashing. Future Generation Computer Systems, 164*, 107586. https://doi.org/10.1016/j.future.2024.107586
63. Huang, Y., Dai, X., Zhang, Q., & Li, Z. (2024). *A new 3D robust chaotic mapping and its application to speech encryption. Chaos, Solitons & Fractals, 184*, 112731. https://doi.org/10.1016/j.chaos.2024.112731
64. Sharobim, B. K., Yacoub, M. H., Said, L. A., & Radwan, A. G. (2024). *Hardware implementation of a 2D chaotic map-based audio encryption system using SBox. Electronics, 13*(21), 4254. https://doi.org/10.3390/electronics13214254
65. Zhang, Q., Huang, Y., & Dai, X. (2025). *Speech signal security scheme based on multiple chaotic maps with deoxyribonucleic acid coding algorithm. International Journal of Electronics and Telecommunications, 71*(2), 157–168. https://doi.org/10.24425/ijet.2025.153546
66. AlOmari, M., & Elmanfaloty, R. (2022). *A combined chaotic system for speech encryption. Engineering, Technology & Applied Science Research, 12*(5), 8641–8650. https://doi.org/10.48084/etasr.4912

67. Al-Rubaie, M., & Abou-Bakr, E. (2022). *Digital speech files encryption based on Hénon and Gingerbread chaotic maps. Iraqi Journal of Science, 63*(4), 1075–1085. https://doi.org/10.24996/ijs.2022.63.4.5
68. Xue, L., Sun, H., Sun, N., Jia, B., & Xiao, D. (2026). *DPSHE: A privacy-preserving federated learning scheme based on homomorphic encryption and differential privacy for medical image. Computers & Electrical Engineering, 129*, 110830. https://doi.org/10.1016/j.compeleceng.2025.110830
69. Puviarasu, A., & Sudha, V. K. (2026). *Enhanced IoT security: Privacy-preserving federated learning model for accurate, real-time intrusion detection across devices. Ain Shams Engineering Journal, 17*(1), 103866. https://doi.org/10.1016/j.asej.2025.103866
70. Chen, Y. (2026). *Decentralized federated learning enables privacy-preserving NIR spectroscopy calibration: A proof-of-concept study. Chemometrics and Intelligent Laboratory Systems, 268*, 105583. https://doi.org/10.1016/j.chemolab.2025.105583
71. Babu, G., Geetha, T. S., Nedumaran, A., Sugumaran, B., & Muralikrishnan, P. (2026). *Integrating quantum computing with federated learning for enhanced security and privacy in IoT networks. Results in Engineering, 29*, 108500. https://doi.org/10.1016/j.rineng.2025.108500
72. Kingma, D. P., & Ba, J. (2015). *Adam: A method for stochastic optimization.* In 3rd International Conference on Learning Representations (ICLR 2015). https://arxiv.org/abs/1412.6980
73. Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., & Han, J. (2020). *On the variance of the adaptive learning rate and beyond.* In International Conference on Learning Representations (ICLR 2020). https://openreview.net/forum?id=r1lYxbRcFQ
74. Wilson, A. C., Roelofs, R., Stern, M., Srebro, N., & Recht, B. (2017). *The marginal value of adaptive gradient methods in machine learning.* In Advances in Neural Information Processing Systems (NeurIPS 2017), 30, 4148–4158. https://arxiv.org/abs/1705.08292
75. Reddi, S. J., Kale, S., & Kumar, S. (2018). *On the convergence of Adam and beyond.* In 6th International Conference on Learning Representations (ICLR 2018). https://arxiv.org/abs/1904.09237
76. Luo, L., Xiong, Y., Liu, Y., & Sun, X. (2019). *Adaptive gradient methods with dynamic bound of learning rate.* In 7th International Conference on Learning Representations (ICLR 2019). https://arxiv.org/abs/1902.09843
77. Wang, T., Blocki, J., Li, N., & Jha, S. (2019). Locally differentially private protocols for frequency estimation. *Proceedings of the IEEE 35th International Conference on Data Engineering (ICDE)*, 729–740. https://doi.org/10.1109/ICDE.2019.00070
78. Bassily, R., Smith, A., & Thakurta, A. (2017). Private empirical risk minimization: Efficient algorithms and tight error bounds. *Proceedings of the IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, 464–475. https://doi.org/10.1109/FOCS.2017.51
79. Rahmati M (2025), "Federated learning for privacy-preserving AI in human–robot collaboration for smart manufacturing". *Journal of Intelligent Manufacturing and Special Equipment*, Vol. 6 No. 2 pp. 210–224, doi: https://doi.org/10.1108/JIMSE-03-2025-0003