Computing and Informatics, Vol. 45, 2026, 1236–1271, doi: 10.31577/cai_2026_5_1236

FL_SELHE: A SELECTIVE HOMOMORPHIC ENCRYPTION-BASED FEDERATED LEARNING FRAMEWORK FOR ROBUST PRIVACY PROTECTION AGAINST GRADIENT LEAKAGE ATTACKS

Mohammed El Aymene Beyat*, Mohammed Kamel Benkaddour

Department of Computer Science and Information Technology
Artificial Intelligence and Information Technology Laboratory (LINATI)
University of Kasdi Merbah, Ouargla, 30000, Algeria
e-mail: {beyat.mohammedelaymene, benkaddour.kamel}@univ-ouargla.dz

Abdessalam Moualdi

Department of Electronics and Telecommunications
Laboratory of Electrical Engineering (LAGE)
Faculty of New Technologies of Information and Communication
University of Kasdi Merbah, Ouargla, 30000, Algeria
e-mail: moualdi.abdessalam@univ-ouargla.dz

Abdelalim Belmesmar, Mohammed El Amine Beyat

Department of Computer Science and Information Technology,
Artificial Intelligence and Information Technology Laboratory (LINATI),
University of Kasdi Merbah, Ouargla, 30000, Algeria
e-mail: {belmesmar.abdelalim, beyat.mohammedelamine}@univ-ouargla.dz

Abstract. Federated learning (FL) promises to protect user data through collaborative training, yet gradient leakage attacks can accurately reconstruct sensitive information from shared updates. While homomorphic encryption (HE) offers a formidable defense, its high computational and communication costs often

* Corresponding author

make it impractical for real-world deployment. In this work, we propose FL_SelHE Federated Selective Homomorphic Encryption, a framework that strategically secures only the most vulnerable layers of a model. By using a one-time sensitivity profiling based on gradient magnitudes, we identify critical layers and apply the CKKS encryption scheme selectively. Experimental evaluation demonstrates that the framework achieves 94.79 % accuracy, comparable to unencrypted training, while reducing the aggregation time by over 40 % compared to fully encrypted models. Furthermore, under gradient inversion attacks, the reconstructed images degrade to near-random noise (PSNR ≈ 5 dB and SSIM ≈ 0), confirming strong resistance to data leakage. These findings prove that selective encryption is a scalable and efficient solution for robust privacy in sensitive domains like healthcare.

Keywords: Federated learning, homomorphic encryption, selective encryption, gradient leakage attacks, CKKS scheme, sensitive layer protection

Mathematics Subject Classification 2010: 68T05, 68P25, 94A60

1 INTRODUCTION

Artificial intelligence has become a foundational technology across numerous high-stakes domains, particularly in healthcare, finance, and personalized services, where intelligent decision-making must coexist with strict regulatory compliance and ethical data stewardship [1, 2]. As deep learning systems increasingly power diagnostic tools, clinical decision support, and predictive analytics, they often rely on large-scale datasets that contain sensitive user information. Traditional centralized machine learning pipelines, which require data to be aggregated on a central server for training, introduce significant privacy and legal risks, particularly under stringent frameworks such as the General Data Protection Regulation (GDPR) and Health Insurance Portability and Accountability Act (HIPAA) [3, 4]. Federated learning (FL) has emerged as a compelling alternative [5], allowing decentralized clients to collaboratively train a global model without transferring raw data. In this framework, each client computes local model updates using its private dataset, and only the resulting model gradients or updated parameters are sent to the central server for aggregation. This design minimizes data exposure and allows for distributed learning within privacy constraints. However, federated learning is not inherently secure. Gradient-sharing mechanisms have been shown to leak sensitive information, with attacks such as Deep Leakage from Gradients (DLG) [6], capable of reconstructing input data only from shared model updates. Such vulnerabilities expose a fundamental gap between privacy intent and the technical reality in federated learning systems.

A variety of privacy-preserving techniques have been proposed to address this gap. Differential privacy adds statistical noise to gradient updates, offering prov-

able privacy guarantees [7] at the expense of utility degradation. Secure multi-party computation (SMPC) enables collaborative operations while maintaining the confidentiality of individual inputs. However, its computational and communication overhead scales poorly with model complexity [8]. Homomorphic encryption (HE), on the other hand, allows computations to be performed directly on encrypted data [9], preserving privacy without affecting model architectures or gradients. Although HE offers a powerful cryptographic guarantee, its integration into practical FL workflows remains challenging due to the significant computational burden it introduces, especially when encrypting high-dimensional deep learning models [10].

To address this limitation, we developed a homomorphically encrypted federated learning framework based on the CKKS scheme, tailored for efficient encrypted computation over real-valued vectors. Unlike traditional approaches that fully encrypt all model parameters, our framework introduces a selective encryption mechanism that targets only the most privacy-sensitive layers. These layers are identified using a one-time sensitivity profiling phase based on gradient magnitude analysis, allowing the system to focus protection where it is most needed. This design reduces the computational load and communication overhead without weakening the model's resistance to gradient-based attacks. The remainder of this paper is organized as follows. Section 2 introduces the necessary background on federated learning and encryption-based privacy preservation. Section 3 outlines the proposed methodology, which includes the system design, encryption strategies, and sensitivity-based selective protection. Section 4 delineates the experimental configuration, including the dataset, evaluation metrics, training parameters, and learning specifications. Section 5 details the experimental results, encompassing centralized benchmarking of candidate architectures, federated learning performance under IID and non-IID distributions, and a thorough evaluation of the encrypted federated framework in terms of accuracy, computational overhead, and resilience to privacy leakage. Section 6 examines system-level trade-offs and their broader implications, whereas Section 7 summarizes the findings and outlines potential directions for future research.

2 BACKGROUND AND RELATED WORK

Privacy-preserving federated learning (FL) has become a fundamental method for facilitating collaborative model training while maintaining the confidentiality of client data. This concept is especially important in sensitive domains like healthcare and finance, where regulatory and ethical constraints prevent centralized data aggregation. Still, conventional FL remains susceptible to privacy threats, including gradient inversion and deep leakage attacks, which exploit model updates to reconstruct private training data. As a result, the research community has focused on developing mechanisms to secure the FL pipeline while maintaining its utility. Among the prevailing lines of defense, three dominant strategies have taken shape: differential privacy (DP), secure multi-party computation (SMPC), and homomorphic encryption (HE).

Differential privacy introduces random noise to model updates, thereby ensuring that individual client contributions remain statistically indistinguishable within the aggregated outcome. Wei et al. [11] proposed a globally differentially private FL framework that optimally balances noise magnitude and model utility using Gaussian perturbation. Wang et al. [12] extended this approach with an adaptive privacy budget allocation mechanism that responds dynamically to the training process, including loss, accuracy, and client-specific data volume. On the client side, Truex et al. [13] introduced LDP-Fed, a local differential privacy scheme that injects noise prior to communication, eliminating reliance on a trusted server. This approach was later improved by Wu and Gao [14], who incorporated the SCAFFOLD optimization method to counteract client drift and preserve convergence under high-noise conditions, particularly valuable in non-IID federated settings.

In parallel, SMPC has been widely adopted to ensure that model updates are aggregated without exposing intermediate values to any party. Zhu et al. [15] demonstrated a privacy-preserving optimization protocol that decouples cryptographic computation from model update operations using oracle-aided SMPC, thereby minimizing latency. Li et al. [16] introduced Chain-PPFL, a serial masking mechanism in which updates pass through clients in sequence, substantially reducing communication complexity. Further advancements by Kanagavelu et al. [17] focused on compressing encrypted gradients to conserve bandwidth while retaining SMPC guarantees. Chen et al. [18] expanded on these ideas by integrating compressed sensing, thereby achieving robustness in larger-scale deployments with sparse communication channels.

While DP and SMPC provide significant protection, homomorphic encryption has gained attention as a particularly effective method because it allows for mathematical operations to be conducted directly on encrypted data. In contrast to DP, which compromises utility for privacy, and secure multi-party computation, which involves significant communication costs, homomorphic encryption maintains data confidentiality throughout the entire computational process without necessitating data decryption at any stage. Zhang et al. [19] proposed BatchCrypt, which integrates leveled CKKS encryption with quantization and batching to improve the efficiency of secure aggregation in cross-silo FL settings. Jayaram et al. [20] developed MYSTIKO, a framework using the Paillier cryptosystem to enable exact encrypted gradient aggregation with minimal trust assumptions. Building upon this, Ma et al. [21] proposed a multi-key homomorphic encryption (MKHE) scheme, which enables each client to use a distinct public key, enhancing scalability and eliminating centralized key sharing. Choosing the right privacy-preserving technique in federated learning is determined by the desired balance of privacy rigor, system performance, and implementation complexity. DP offers formal guarantees by introducing statistical noise, with Central DP providing better utility under the assumption of a trusted server [11], while Local DP ensures client-side autonomy at the cost of model performance [22]. DP's lightweight implementation makes it ideal for mobile or edge devices, although noise must be carefully tuned to maintain accuracy. Secure Multi-Party Computation provides privacy through cryptographic

protocols while maintaining model fidelity [23]. SMPC ensures exact computation without data leakage, but it incurs high communication overhead and synchronization complexity, especially in asynchronous or large-scale deployments [24].

Homomorphic Encryption (HE) balances data confidentiality with model accuracy by allowing encrypted computations. Fully or leveled HE schemes like CKKS support aggregation over ciphertexts without decryption, enabling secure training on sensitive data [19]. While HE introduces higher computational costs and large ciphertexts, recent advancements such as selective parameter encryption and multi-key designs greatly reduce the overhead [25]. Thus, HE is gaining preference in privacy-sensitive sectors such as healthcare and finance.

3 PROPOSED METHODOLOGY

3.1 System Overview

To tackle the critical issue of privacy leakage in federated learning, we propose an improved framework that incorporates homomorphic encryption (HE) into the standard FL pipeline. The goal is to secure sensitive client-side information not only at the data level, but also during model update exchange and aggregation. Our design assumes a semi-honest threat model in which the server adheres to the protocol but may attempt to extract private information from received updates. We believe that the choice of a **semi-honest (honest-but-curious) threat model** is not only practical but also appropriate for the cross-silo scenarios we address. In sensitive domains like healthcare or finance, participants are usually established institutions (e.g., hospitals) bound by strict legal agreements and ethical codes. While they are unlikely to maliciously tamper with the training process, the passive extraction of patient data from shared gradients remains a very real threat. By focusing on this model, we can implement high-level protection like CKKS without the extreme computational overhead that a "malicious" model (requiring complex Zero-Knowledge Proofs) would demand, making it a more realistic solution for actual medical applications. The workflow of the proposed framework is organized as follows:

1. The central server creates a global model and distributes it to each client.
2. Each client trains the model locally on its private dataset.
3. After completing local training, clients encrypt their model updates using a homomorphic encryption scheme.
4. The encrypted updates are transmitted to the server and aggregated directly on ciphertexts without decryption.
5. The aggregated encrypted model is sent back to clients, who decrypt and continue the next training round.

This encryption-based orchestration ensures end-to-end confidentiality of model updates, reducing the risk of gradient leakage and adversarial inference. The pri-

mary technical contribution is the incorporation of selective homomorphic encryption, which is applied only to the most privacy-sensitive layers as identified empirically through gradient sensitivity profiling. By limiting encryption to these layers, the framework reduces computational overhead while remaining highly resistant to inversion attacks. This targeted strategy provides a practical and scalable solution to the ongoing tension between privacy and efficiency in real-world FL deployments. Figure 1 summarizes the proposed workflow.

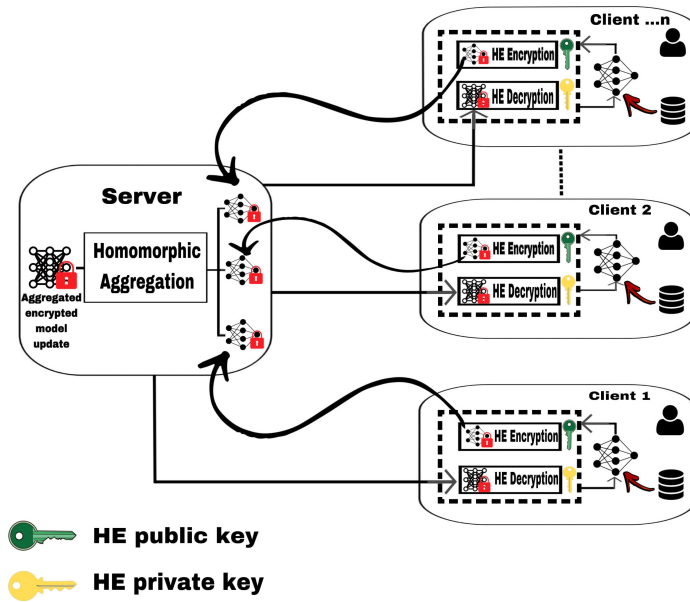


Figure 1. Overview of the proposed homomorphic encryption-enhanced federated learning framework

3.2 Homomorphic Encryption Strategy

To protect sensitive model updates during federated learning, the proposed framework implements two distinct homomorphic encryption strategies: full and selective encryption. While the full encryption strategy maximizes privacy by securing all model parameters, the selective encryption strategy is intended to target only the most vulnerable components, typically layers with high gradient sensitivity, providing a computationally efficient alternative while maintaining data security. At the heart of these strategies lies the Cheon–Kim–Kim–Song (CKKS) scheme [26], a leveled homomorphic encryption method built on the Ring Learning With Errors (RLWE) assumption.

3.2.1 The Cheon–Kim–Kim–Song (CKKS) Scheme

The CKKS scheme, introduced by Cheon et al. [26], is a leveled homomorphic encryption method based on the Ring Learning With Errors (RLWE) assumption. Its key feature is support for approximate arithmetic on vectors of complex numbers, enabling secure computations on encrypted data while tolerating small numerical errors, a trade-off well suited to deep learning where exact precision is unnecessary. CKKS operates over polynomial rings with ciphertexts in $R_q = \mathbb{Z}_q[X]/(X^N + 1)$, where N is a power of two and q is a large modulus. Messages are vectors in $\mathbb{C}^{N/2}$ encoded into polynomials. The main stages are:

Key generation: a secret key $s(X)$ is sampled; the public key incorporates random polynomials and noise to ensure security.

Encoding and encryption: input vectors are scaled and mapped into R_q, then encrypted into ciphertext pairs (c_0, c_1).

Homomorphic operations: addition and multiplication are performed directly on ciphertexts, with relinearization and rescaling controlling ciphertext size and precision.

Decryption and decoding: ciphertexts are decrypted with the secret key, rescaled, and projected back to approximate the original vector.

3.2.2 Full Homomorphic Encryption Strategy

In this strategy, each client encrypts all model parameters with the CKKS scheme before transmission, ensuring confidentiality of updates during communication and aggregation. The overall workflow is shown in Algorithm 1, while decryption, encryption, and secure aggregation are detailed in Algorithms 2, 3, and 4.

DecryptModel (Algorithm 2) restores the global model to its tensor form for local training, while EncryptModel (Algorithm 3) reverses the process before updates are returned. The server then applies EncryptedFedAvg (Algorithm 4) to aggregate them via weighted summation in the ciphertext domain.

3.3 Sensitivity Analysis and Layer Selection

Before selective encryption, our framework performs a one-time sensitivity-profiling step to identify layers most vulnerable to gradient leakage, guided by the ℓ_2-norm of gradient magnitudes observed on mini-batches. The process combines two stages:

1. manually flagging layers with known sensitivity (e.g., classifier layers), and
2. selecting additional layers with the highest averaged gradient norms.

This hybrid approach captures both empirical and structural sensitivity. The complete procedure is given in Algorithm 5.

Algorithm 1 Federated Learning with Full Homomorphic Encryption

Server executes:

Initialize global model W_0, total rounds T
for $t \leftarrow 1$ to T **do**
  Select $m = \max(\lfloor \rho \cdot K \rfloor, 1)$ clients as $\mathcal{S}_t$
  $\mathcal{U}_t \leftarrow \emptyset$ ▷ Encrypted updates
  for all $c \in \mathcal{S}_t$ **in parallel do**
    $C_{t+1}^{(c)}, n_c \leftarrow$ **ClientUpdate**(c, W_t)
    Append $\left(C_{t+1}^{(c)}, n_c\right)$ to $\mathcal{U}_t$
  end for
  $C_{t+1}^{\text{agg}} \leftarrow$ **EncryptedFedAvg**$(\mathcal{U}_t, Pk)$ ▷ Algorithm 4
  $W_{t+1} \leftarrow C_{t+1}^{\text{agg}}$ and broadcast to clients
end for

ClientUpdate(c, W_t)**:**

Decrypt W_t with Sk using **DecryptModel** (Algorithm 2)
Train on local data using SGD
Encrypt updated model $W_{t+1}^{(c)}$ using **EncryptModel** (Algorithm 3)
return $\left(C_{t+1}^{(c)}, n_c\right)$

Algorithm 2 DecryptModel

Require: Encrypted parameters $\mathcal{C} = \{c_1, \ldots, c_n\}$, secret key Sk, shapes $\mathcal{S}$
Ensure: Plaintext model $\mathcal{P}$

$\mathcal{P} \leftarrow \emptyset$
for $i \leftarrow 1$ to n **do**
  $\tilde{p}_i \leftarrow \text{CKKS_Decrypt}(Sk, c_i)$
  $p_i \leftarrow \text{Reshape}(\tilde{p}_i, s_i)$
  Append p_i to $\mathcal{P}$
end for
return $\mathcal{P}$

Algorithm 3 EncryptModel

Require: Plaintext parameters $\mathcal{P}$, public key Pk
Ensure: Encrypted parameters $\mathcal{C}$

$\mathcal{C} \leftarrow \emptyset$
for $i \leftarrow 1$ to n **do**
  $p_i^{\text{flat}} \leftarrow \text{Flatten}(p_i)$
  $c_i \leftarrow \text{CKKS_Encrypt}\left(Pk, p_i^{\text{flat}}\right)$
  Append c_i to $\mathcal{C}$
end for
return $\mathcal{C}$

Algorithm 4 EncryptedFedAvg

Require: Client updates $\left\{\left(\mathcal{C}^{(j)}, n_j\right)\right\}_{j=1}^{m}$, public key Pk
Ensure: Aggregated encrypted model $\mathcal{C}^{\text{agg}}$
$N \leftarrow \sum_{j=1}^{m} n_j$
$\mathcal{C}^{\text{agg}} \leftarrow \emptyset$
for $i \leftarrow 1$ to n **do**
  $c_i^{\text{agg}} \leftarrow 0$
  for $j \leftarrow 1$ to m **do**
    $w \leftarrow \frac{n_j}{N}$
    $c_i^{\text{agg}} \leftarrow c_i^{\text{agg}} + \left(\mathcal{C}^{(j)}[i] \cdot w\right)$
  end for
  Append c_i^{agg} to $\mathcal{C}^{\text{agg}}$
end for
return $\mathcal{C}^{\text{agg}}$

Algorithm 5 Determine_Sensitivity

Require: Model M, dataset $\mathcal{D}$, batch size m, number of batches T, mandatory prefixes $\mathcal{P}$, top fraction γ
Ensure: Annotated list $\mathcal{I} = \{(name_l, shape_l, sensitive_l)\}_{l=1}^{L}$
$\mathcal{I} \leftarrow \emptyset$, $\mathcal{N} \leftarrow \emptyset$
for $l \leftarrow 1$ to L **do**
  Extract $name_l$, $shape_l$, initialize $is_mandatory \leftarrow$ False
  for all $prefix \in \mathcal{P}$ **do**
    if $name_l$ starts with $prefix$ **then**
      Append $(name_l, shape_l, \textbf{True})$ to $\mathcal{I}$; $is_mandatory \leftarrow$ True
    end if
  end for
  if not $is_mandatory$ **then**
    $total_norm \leftarrow 0$
    for $t \leftarrow 1$ to T **do**
      Sample batch B_t from $\mathcal{D}$, compute gradient $\nabla_{\theta_l}\mathcal{L}(B_t)$
      $total_norm \mathrel{+}= \|\nabla_{\theta_l}\mathcal{L}(B_t)\|_2$
    end for
    $\bar{G}_l \leftarrow total_norm/T$, Append $(l, \bar{G}_l)$ to $\mathcal{N}$
  end if
end for
Sort $\mathcal{N}$ by $\bar{G}_l$ descending; select top-γ % as sensitive
for $l \leftarrow 1$ to L **do**
  if not already in $\mathcal{I}$ **then**
    Append $(name_l, shape_l, \textbf{True})$ if l in top-k; else **False**
  end if
end for
return $\mathcal{I}$

3.4 Selective Encryption Strategy

The selective encryption strategy increases system efficiency by encrypting only sensitive layers, thereby reducing ciphertext size and encryption overhead while ensuring privacy. This targeted approach enables scalability in federated environments without compromising security. The selective-encryption algorithms mirror their full-encryption counterparts in structure, with necessary adaptations to support partial encryption. Parameters are encrypted, decrypted, and aggregated via the hybrid mechanism in Algorithm 6.

A one-time sensitivity analysis is performed at the beginning of training to determine the encryption ratio. Gradients are monitored to identify the most privacy-sensitive areas, which are usually located in the final layers of client-specific decision boundaries. Empirical profiling consistently identified the top 15 % of parameters as the optimal cutoff, balancing resistance to model inversion and gradient leakage with computational and communication constraints. This threshold allocates cryptographic protection to areas of greatest need, ensuring security and scalability.

Algorithm 6 Selective_Encrypted_FedAvg

Require: Submissions $\left\{\left(\mathcal{C}^{(j)}, n_j\right)\right\}_{j=1}^{m}$, info list $\mathcal{I}$
Ensure: Aggregated model $\mathcal{C}^{\text{agg}}$
$N \leftarrow \sum_{j=1}^{m} n_j$, $\mathcal{C}^{\text{agg}} \leftarrow \emptyset$
for $i \leftarrow 1$ to number of parameters **do**
  $(_, _, s_i) \leftarrow \mathcal{I}[i]$, $c_i^{\text{agg}} \leftarrow 0$
  for $j \leftarrow 1$ to m **do**
    $w_j \leftarrow \frac{n_j}{N}$, $val \leftarrow \mathcal{C}^{(j)}[i]$
    $c_i^{\text{agg}} \leftarrow c_i^{\text{agg}} + (val \cdot w_j)$
  end for
  Append c_i^{agg} to $\mathcal{C}^{\text{agg}}$
end for
return $\mathcal{C}^{\text{agg}}$

3.5 Security Analysis and Comparison with SMPC

To better understand the positioning of FL_SelHE, it is helpful to compare it with other Privacy-Preserving Techniques (PPTs).

Secure Multi-Party Computation (SMPC): SMPC protocols provide high security by ensuring no party sees individual updates. However, they are highly interactive and require massive communication bandwidth, which is often a bottleneck for large models like our EfficientNet-B0 (5.3 M parameters).

Hybrid Solutions: Some methods combine SMPC and HE. While very secure, they significantly increase implementation complexity.

3.6 Experimental Environment

3.6.1 Hardware and Execution Platform

All experiments were conducted using Google Colab [27], a cloud-based environment equipped with an NVIDIA L4 GPU (22.5 GB VRAM), an Intel Xeon CPU, and approximately 53 GB of RAM. This configuration provided a scalable and reliable platform for evaluating both deep learning and homomorphic encryption tasks.

3.6.2 Software Stack

The framework was implemented in Python using widely adopted machine learning libraries. Federated orchestration was managed via Flower [28], while homomorphic encryption functionality was integrated using TenSEAL [29], which enabled CKKS-based secure tensor operations directly within PyTorch models.

4 EXPERIMENTAL SETUP

4.1 Dataset Description

Given the privacy-sensitive nature of healthcare data, our evaluation focuses on medical image classification. We employ the PathMNIST subset from the MedMNIST v2 benchmark [30], a standardized collection designed for lightweight, reproducible experimentation in biomedical imaging.

PathMNIST is derived from colorectal histopathology datasets (NCT-CRC-HE-100K and CRC-VAL-HE-7K) and contains 107 180 RGB image patches resized to 28×28 pixels. The classification task involves nine tissue types, constituting a 9-class classification problem. The dataset comprises 89 996 training, 10 004 validation, and 7 180 test samples. An overview is presented in Table 1. Representative samples and the class distribution are shown in Figures 2 and 3, respectively.

Aspect	Details
Modality	Colon Histopathology
Total Samples	107 180
Training/Validation/Test	89 996/10 004/7 180
Image Format	RGB, 28×28 pixels
Classes	9 tissue types
Task	Multi-class classification

Table 1. PathMNIST dataset overview

To prepare the data for CNN-based training, we applied a standard preprocessing pipeline including:

Tensor Conversion: Image data was converted to tensor format for efficient GPU processing.

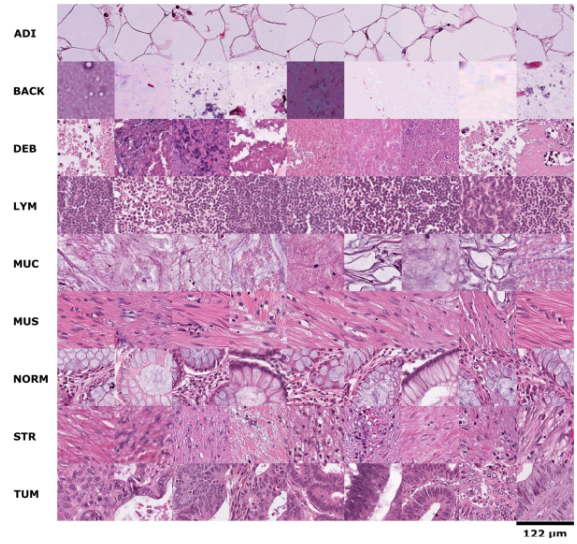


Figure 2. Example images from each of the nine tissue classes in PathMNIST

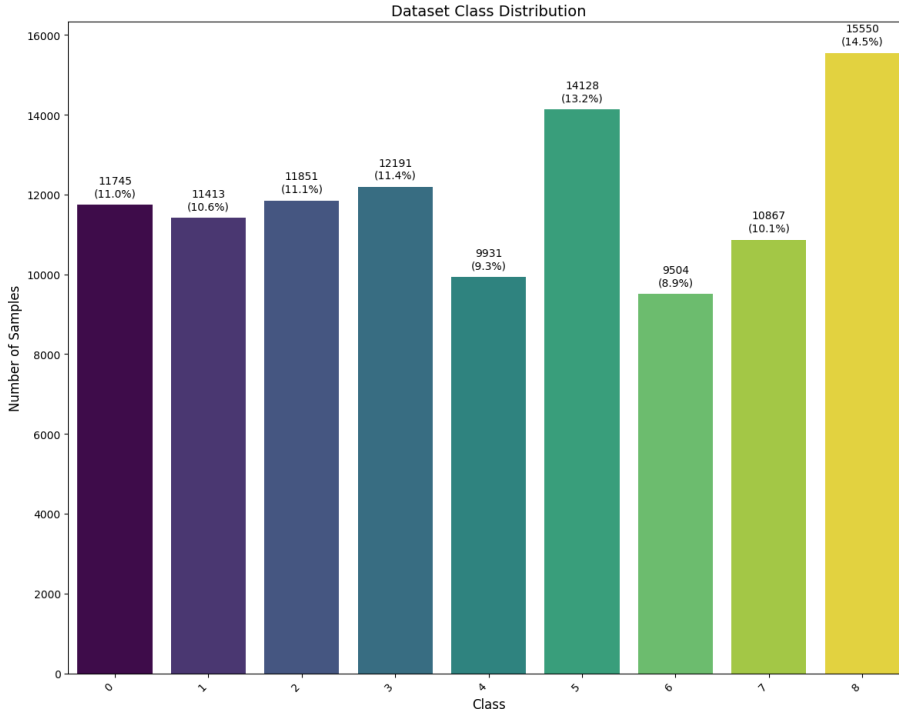


Figure 3. Class distribution in the PathMNIST dataset

Normalization: Pixel intensities were normalized to stabilize training.

Data Augmentation: Random flips, rotations, and crops were applied during training to improve generalization.

To evaluate the proposed framework under different configurations (centralized, federated, and encrypted), we employ a comprehensive set of performance and overhead metrics. These include standard classification scores, computational costs, and privacy leakage estimations.

4.2 Classification Performance

We report accuracy [31], precision [32], recall [33], and F1-score [34] to assess predictive performance across all experimental configurations. These metrics, derived from multiclass confusion matrices, provide a balanced perspective on classification effectiveness, especially in the presence of class imbalance.

4.3 Computational Overhead

To quantify the additional costs introduced by homomorphic encryption and federated learning, we measure the following:

Training Time: Duration of local model updates.

Encryption/Decryption Time: Time consumed securing model updates via HE.

Aggregation Time: Time taken to perform secure aggregation on the server.

(De)Serialization Time: Overhead from converting encrypted vectors to/from transmittable formats.

Total Round Time: End-to-end time per communication round.

4.4 Privacy Leakage Metrics

To evaluate resistance to gradient inversion attacks, we adopt two image similarity metrics commonly used in reconstruction analysis:

4.4.1 Peak Signal-to-Noise Ratio (PSNR)

PSNR quantifies pixel-level similarity between original and reconstructed images:

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{\text{MSE}} \right). \tag{1}$$

Higher PSNR values indicate better visual fidelity and, consequently, a greater privacy leakage risk.

4.4.2 Structural Similarity Index (SSIM)

SSIM captures perceptual similarity based on luminance, contrast, and structural features:

$$\mathrm{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}. \tag{2}$$

Values closer to 1 indicate higher similarity and leakage, while lower values indicate stronger privacy preservation.

5 EXPERIMENTS AND RESULTS

5.1 Centralized Benchmarking of Deep Learning Architectures

Before deploying our privacy-preserving federated learning framework, we carried out extensive benchmarking with centralized deep learning models to determine the most performant architecture for downstream evaluation. This preliminary stage was essential for establishing a high-quality baseline and ensuring a fair comparison against federated and encrypted counterparts. We evaluated both custom-designed convolutional neural networks and state-of-the-art pre-trained models on the PathMNIST dataset. The models considered include CNNMed (with and without self-attention enhancement), ResNet-18 [35], MobileNetV3 Small [36], and EfficientNet-B0 [37].

5.1.1 Custom CNN Architectures

CNNMed is a lightweight convolutional neural network designed for histopathological image classification. It consists of three convolutional layers interspersed with batch normalisation, ReLU activations, and max pooling operations. Dropout and adaptive average pooling are applied to ensure regularisation and feature dimension consistency. The classification head consists of two fully connected layers, followed by a 9-class output layer.

An improved variant, CNNMed with Self-Attention, incorporates a spatial attention mechanism to improve feature extraction. This addition selectively emphasises informative regions in the image in order to improve classification accuracy while incurring minimal computational overhead.

5.1.2 Pre-Trained Architectures

To leverage prior knowledge embedded in large-scale image models, we fine-tuned three well-established architectures:

ResNet-18: A residual network that mitigates vanishing gradients via identity shortcuts, enabling deeper learning with stable convergence.

MobileNetV3 Small: A resource-efficient model designed for low-power devices, combining neural architecture search with advanced activation and attention mechanisms.

EfficientNet-B0: A highly optimized model employing compound scaling to balance depth, width, and resolution, offering superior accuracy-efficiency trade-offs.

These models were selected for their proven generalization capabilities and varying complexity levels, making them suitable for comparative analysis in both centralized and federated contexts.

5.1.3 Results of Centralized Architectures

All models were trained using the same optimisation parameters: a batch size of 64, an initial learning rate of 0.001, and up to 30 training epochs (reduced to 10 for larger pre-trained networks to avoid overfitting). The evaluation employed standard classification metrics precision, recall, F1-score, and overall accuracy averaged across the test set. The resulting comparison is summarized in Table 2 and Figure 4.

Model	Ep.	Prec. (%)	Rec. (%)	F1 (%)	Acc. (%)
CNNMed	30	93.57	93.55	93.33	93.55
+ Self-Attn.	30	94.00	94.11	93.94	94.11
MobileNetV3	10	94.83	94.85	94.82	94.85
ResNet-18	10	95.15	95.04	94.91	95.04
EfficientNet-B0	10	**95.38**	**95.35**	**95.23**	**95.35**

Table 2. Comparison of centralized deep learning models

5.1.4 Selected Baseline: EfficientNet-B0

EfficientNet-B0 outperformed all other evaluated model architectures. It struck an optimal balance between architectural complexity and predictive accuracy, outperforming deeper networks such as ResNet-18 while requiring less computational power than comparable alternatives.

As a result, EfficientNet-B0 was selected as the baseline architecture for all subsequent experiments in both federated learning and privacy-preserving environments. Its robust standalone performance ensures that any trade-offs observed under federated or encrypted settings can be attributed to methodological factors rather than architectural constraints. The corresponding prediction pattern is illustrated by the confusion matrix in Figure 5.

5.2 Federated Learning under IID and Non-IID Distributions

To rigorously evaluate the performance and generalization capabilities of our privacy preserving framework, we implemented and assessed the Federated Learning (FL)

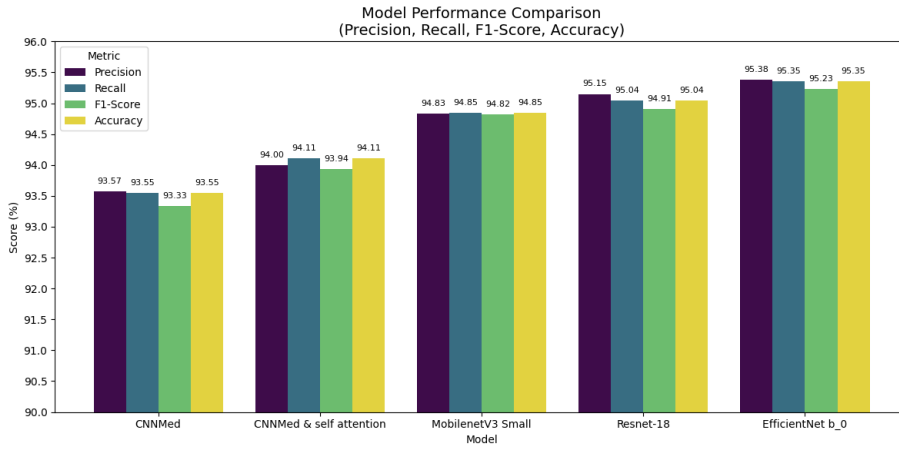


Figure 4. Model performance comparison (Accuracy, Precision, Recall, and F1-score)

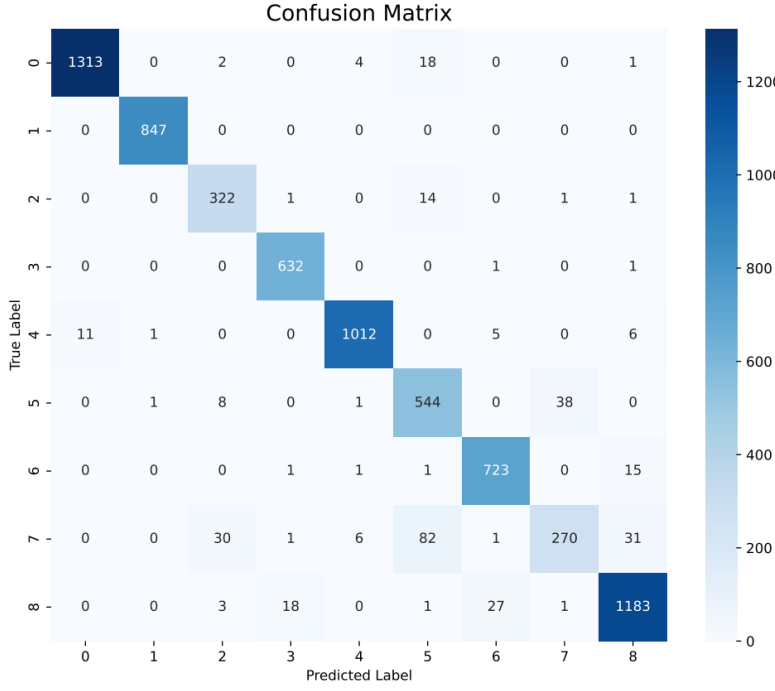


Figure 5. Confusion matrix for EfficientNet-B0 predictions on the PathMNIST

paradigm across two fundamental data distribution regimes: independent and identically distributed (IID) and non-IID. These experiments serve not only to benchmark performance against centralized baselines but also to assess resilience under statistical heterogeneity, a critical real-world concern in decentralized healthcare settings. We adopt the EfficientNet-B0 model as the client-side architecture, based on prior centralized benchmarking. Two federation scales were explored: 5 and 10 clients, to examine the influence of the number of clients on convergence dynamics and predictive performance. All training was performed using the Federated Averaging (FedAvg) algorithm, with optimisation settings summarised in Table 3.

Hyperparameter Settings	
Client Model Architecture	EfficientNet-B0
Number of Clients	5 and 10 (evaluated separately)
Data Distribution Types	IID, Non-IID (Dirichlet $\alpha = 0.5$)
Communication Rounds	10
Local Epochs per Round	1
Local Batch Size	64
Initial Learning Rate	0.001
Aggregation Strategy	FedAvg

Table 3. Federated learning configuration

5.2.1 IID Distribution: Uniform Statistical Balance

In the IID scenario, each client receives a balanced subset of data, ensuring that the class distribution across clients closely resembles that of the global dataset. This setup offers a straightforward way to examine how decentralized optimization behaves in isolation, without the complications introduced by uneven data across clients. Figure 6 illustrates the class-wise sample allocation across five clients under IID distribution. As observed, the statistical parity ensures uniform class exposure, minimizing the variance across local objectives.

Training was run separately with 5 and 10 clients. Figure 7 shows the models' accuracy and loss curves during training. Both setups showed steady progress, indicating stable training and reliable aggregation of model updates despite distributed execution across clients. The performance degradation with increasing client count was marginal, confirming the approach's scalability under uniform data distribution.

Model Configuration	**Accuracy (%)**	**Loss**
Centralized (EfficientNet-B0)	**95.35**	0.1701
Federated (5 Clients)	95.29	**0.1651**
Federated (10 Clients)	94.99	0.1595

Table 4. Centralized vs. federated accuracy and loss (IID)

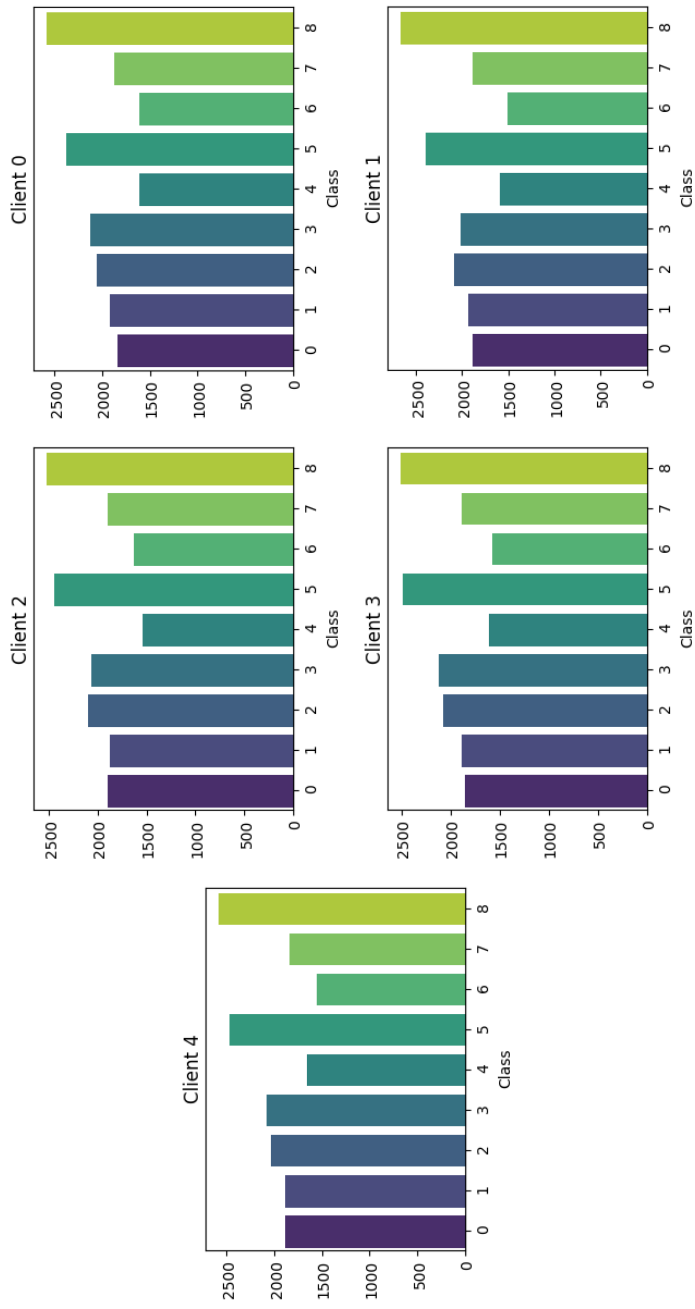


Figure 6. IID data distribution across 5 clients

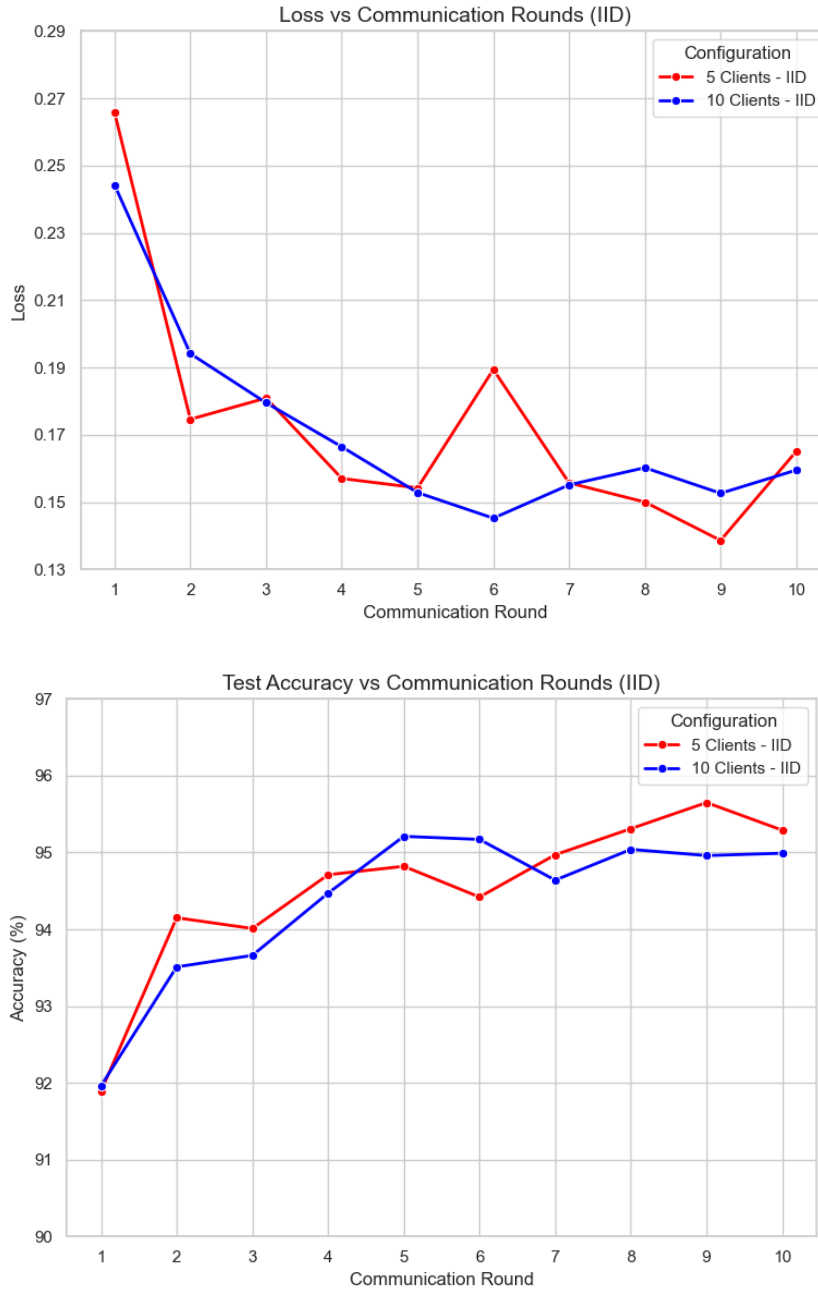


Figure 7. Federated learning accuracy and loss curves (IID)

As summarized in Table 4, even though the training was distributed across different clients, the federated models reached almost the same accuracy as the centralized ones and, in some cases, exhibited slightly lower loss. This shows that when data is evenly balanced across clients (IID), federated learning can come very close to matching centralized results while keeping the data on each client.

5.2.2 Non-IID Distribution: Real-World Statistical Heterogeneity

To reflect how data is typically distributed in real-world situations, we used a Dirichlet-based non-IID split with $\alpha = 0.5$. This is a widely used setting that introduces a fair amount of variation in the data each client sees. In this regime, clients receive imbalanced subsets, with some classes either over- or underrepresented, mimicking institutional biases often seen in federated healthcare environments.

Figure 8 depicts the class-wise imbalance among five clients with non-IID allocations. The resulting skew presents a significant optimization challenge, known as client drift, in which local updates diverge from the global optimum.

The learning curves for 5 and 10 client configurations under non-IID are shown in Figure 9. Compared to the IID case, training becomes noticeably less stable. The models tend to fluctuate more and take longer to settle, especially when using 10 clients. This highlights how difficult it is to combine updates when each client is working with very different data.

Data Type	**Clients**	**Accuracy (%)**	**Loss**
IID	5	95.29	0.1651
IID	10	94.99	0.1595
Non-IID	5	94.26	0.1774
Non-IID	10	93.20	0.2055

Table 5. Federated performance: IID vs non-IID comparison

As summarized in Table 5, although performance under non-IID conditions remains competitive, a measurable drop in accuracy and increased loss is observed compared to IID counterparts. The decline is more pronounced with 10 clients, reinforcing the compounded effect of both statistical heterogeneity and reduced client sampling per round. These findings highlight the critical need for future work on personalization and advanced aggregation techniques (such as FedProx and FedNova) that can mitigate the negative effects of client drift.

5.3 Federated Learning-Based Homomorphic Encryption Framework

This section provides a thorough evaluation of our proposed federated learning framework enhanced with Homomorphic Encryption (HE). The objectives are threefold:

1. to quantify any performance impact introduced by encryption,

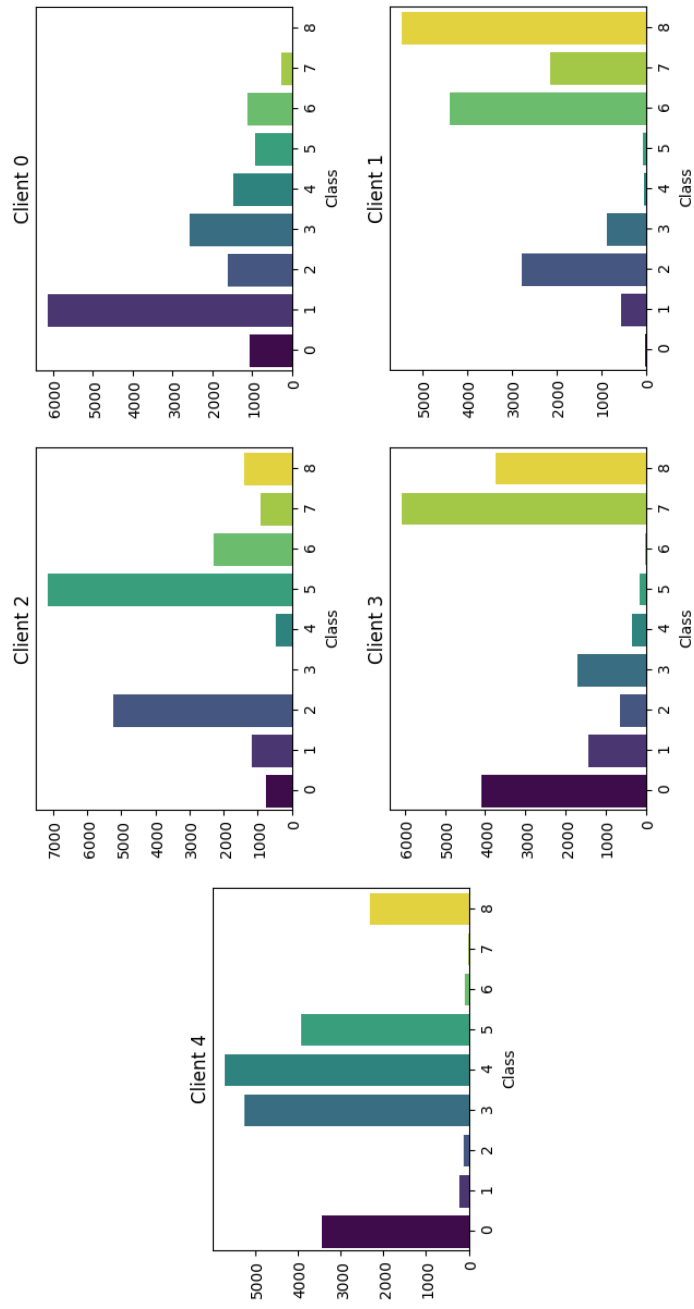


Figure 8. Non-IID data distribution across 5 clients

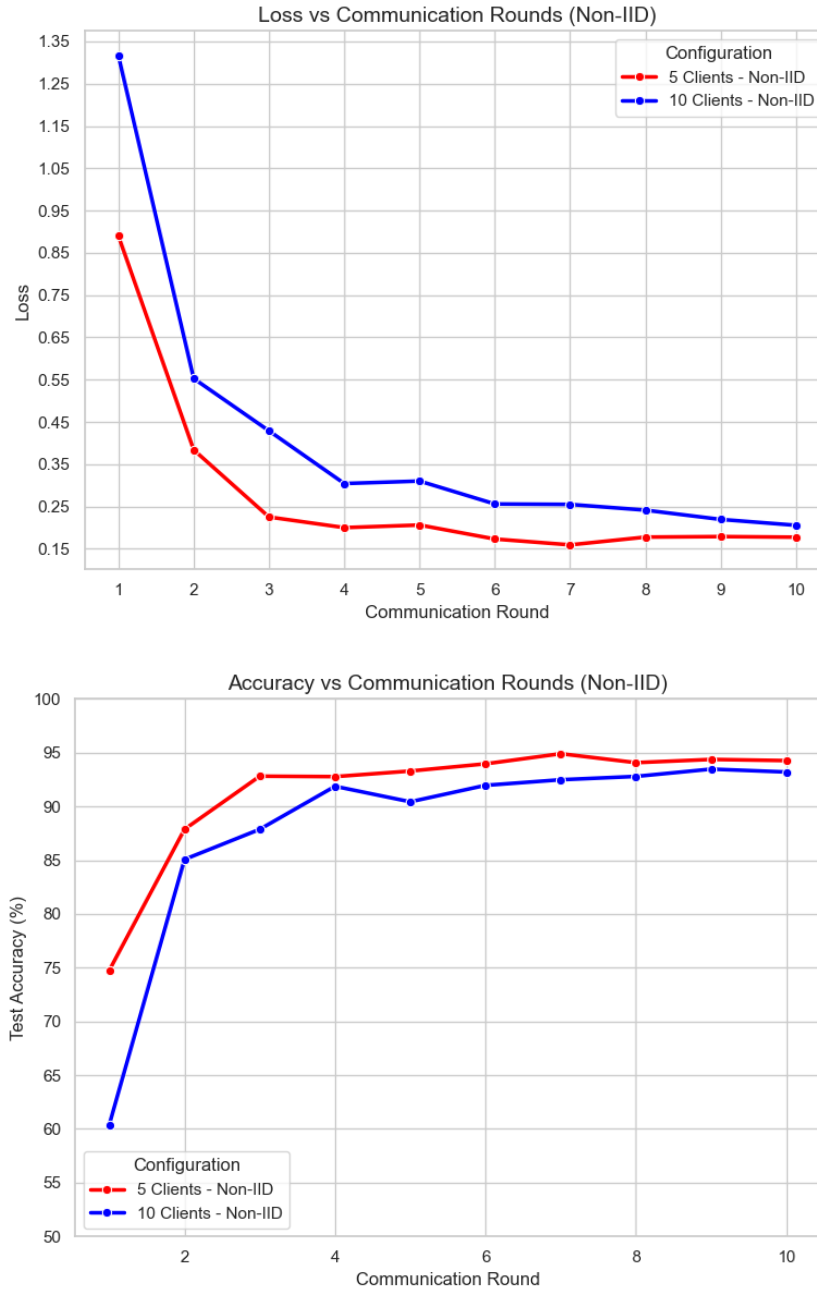


Figure 9. Federated learning accuracy and loss curves (non-IID)

2. to analyze the computational overhead incurred under full and selective HE configurations, and
3. to assess the framework's resilience to privacy attacks, particularly Deep Leakage from Gradients (DLG).

To ensure a focused and interpretable analysis, all experiments in this section were conducted under a standardized IID data distribution with 5 clients. This configuration reduces confounding factors caused by statistical heterogeneity, allowing us to isolate the effects of encryption schemes on model behavior and system performance.

5.3.1 CKKS Encryption and Selective Parameter Protection

The CKKS scheme was chosen for its ability to perform approximate arithmetic on real-valued data, making it particularly well-suited to deep learning applications. All encrypted operations, including aggregation and selective parameter sharing, were performed using the following encryption parameters:

- Polynomial modulus degree: 16 384, enabling the encryption of large parameter vectors with 128-bit security guarantees.
- Coefficient modulus sizes: [60, 40, 40, 60], allowing multi-hop encrypted computations with controlled noise growth.
- Global scale: 2^{40}, which balances precision and ciphertext expansion during encoded computations.

In the selective encryption strategy, we set the encryption ratio to $\gamma = 0.15$, meaning only the top 15 % of model parameters based on gradient sensitivity were encrypted. This threshold was chosen deliberately to balance privacy protection with computational efficiency. Early tests showed that a small group of parameters, especially in the final fully connected and attention layers, consistently had much larger gradient values, suggesting they play a bigger role in how well the model learns and how sensitive it is during predictions. Encrypting these parameters provides robust privacy protection, as they contain the most semantically informative updates, while circumventing the high computational costs linked to encrypting entire models.

Empirical evaluations indicated that raising the encryption ratio beyond 15 % resulted in diminishing returns in terms of privacy benefits, while greatly increasing encryption latency and memory overhead. In contrast, reducing the ratio below this threshold led to significant model drift under adversarial conditions. We found that encrypting 15 % of the most important gradients struck the right balance between maintaining training effectiveness and ensuring scalability. This selective setup is a key part of the framework, offering strong privacy through the CKKS scheme while remaining fast enough for clients with limited computing power.

5.3.2 Model Performance

The core metric for evaluating the practical utility of HE integration is test performance. We compared three configurations:

1. Baseline FL: Federated learning without encryption.
2. Full HE: All shared model updates encrypted.
3. Selective HE: Only a 15 % subset of parameters encrypted.

Method	**Test Accuracy (%)**	**Final Loss**
Baseline FL	95.29	0.165
Full HE (FL_FullHE)	95.13	0.161
Selective HE (FL_SelHE)	94.79	0.167

Table 6. Final model performance (accuracy and loss)

As shown in Table 6 and Figure 10, the impact of encryption on model performance is minimal. Full encryption preserves accuracy within 0.16 % of the baseline, and selective encryption incurs only a marginal additional drop. This shows that CKKS's approximation errors do not substantially degrade learning, and selective protection offers a good balance between privacy and utility.

5.3.3 Computational Overhead Analysis

The integration of HE introduces non-trivial computational costs, which must be evaluated to understand the practical deployment feasibility. We examine both overall round time and client-side execution breakdown for the three configurations.

As observed in Table 7 and Figures 11 and 12, full HE nearly doubles the total round duration due to encryption, decryption, and packing costs. In contrast, selective encryption reduces this overhead by approximately 29 %, offering a more practical deployment path for privacy-preserving FL in real-world systems.

Metric (s)	**FL**	**FL_ FullHE**	**FL_ SelHE**	**↑ FL → FullHE %**	**↑ FL → SelHE %**	**↓ FullHE → SelHE %**
Training	53.98	54.04	54.02	∼ 0 %	∼ 0 %	∼ 0 %
Encryption	N/A	11.75	6.98	N/A	N/A	−40.56 %
Decryption	N/A	1.33	0.86	N/A	N/A	−35.38 %
Serialization	N/A	5.57	3.16	N/A	N/A	−43.25 %
Deserialization	N/A	1.16	0.90	N/A	N/A	−22.30 %
Client Execution	53.98	73.86	65.93	+36.79 %	+22.11 %	−10.74 %
Total Round Time	272.21	403.14	352.02	+48.08 %	+29.30 %	−12.67 %

Table 7. Execution time comparison across FL variants

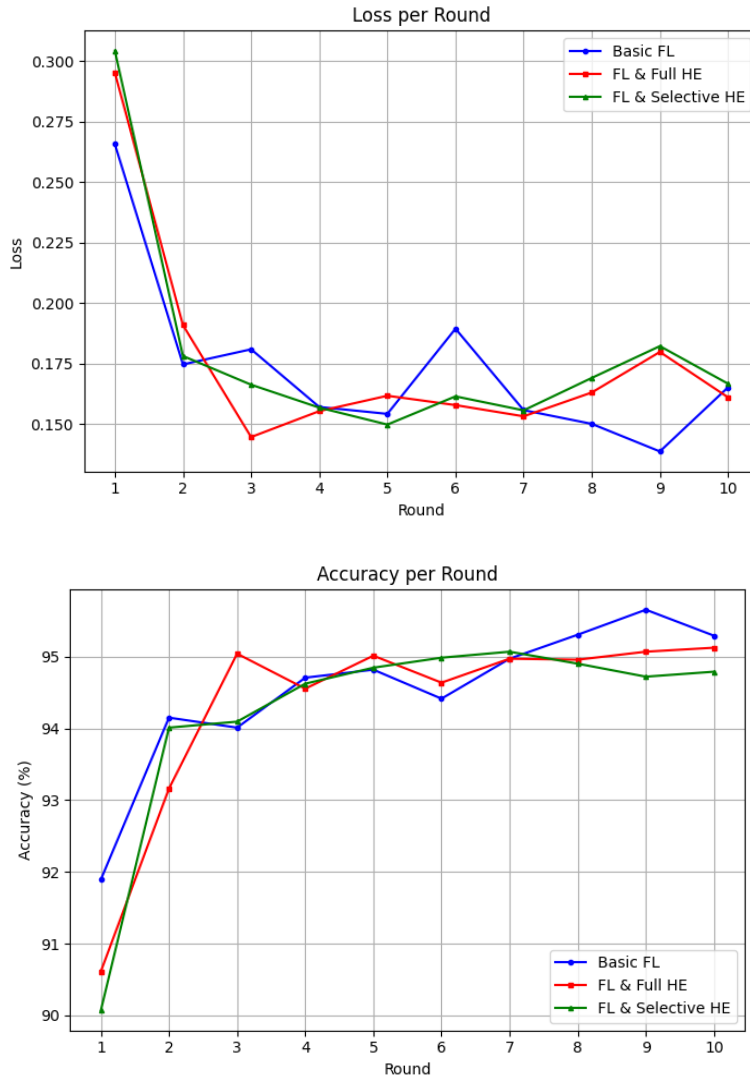


Figure 10. Accuracy and loss across training rounds for FL, FL_FullHE, and FL_SelHE

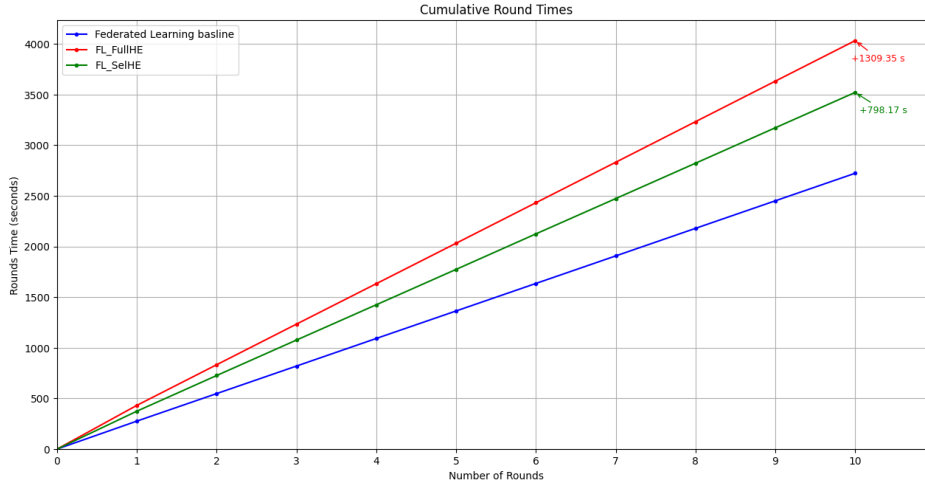


Figure 11. Cumulative round time per training round under FL, FL_FullHE, and FL_SelHE

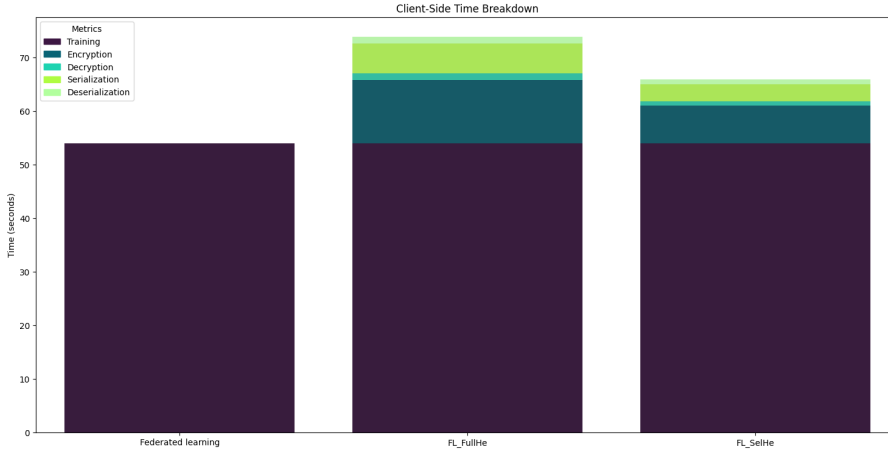


Figure 12. Breakdown of client-side execution time per configuration

5.3.4 Privacy Evaluation via DLG Attack

To verify the robustness of selective encryption against gradient-based leakage, we simulated the Deep Leakage from Gradients (DLG) attack [6, 38]. The attacker attempts to reconstruct private input images from shared gradients during training. The attack pipeline is described in Algorithm 7.

Algorithm 7 Deep Leakage from Gradients (DLG)

Require: Model $\mathcal{F}$ with parameters W; shared gradient ∇W
Ensure: Reconstructed input x, y
1: Initialize $x', y' \sim \mathcal{N}(0, 1)$
2: **for** $i = 1$ to n **do**
3: Compute dummy gradients: $\nabla W_i' \leftarrow \frac{\partial \ell(\mathcal{F}(x_i'), y_i')}{\partial W}$
4: Compute distance: $D_i \leftarrow ||\nabla W_i' - \nabla W||^2$
5: Update x', y' via gradient descent on D_i
6: **end for**
7: **return** x', y'

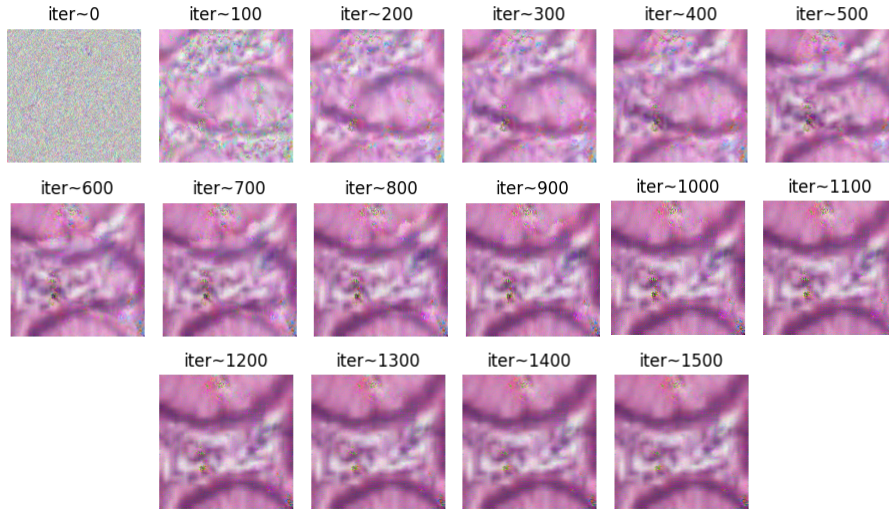


Figure 13. Image reconstruction progression (FL)

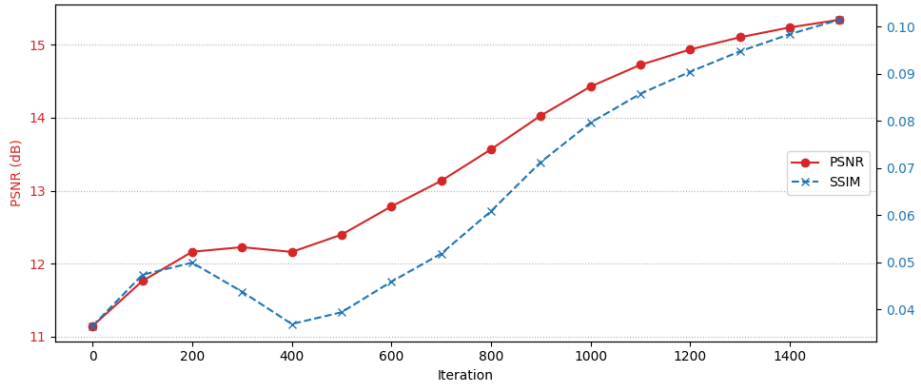


Figure 14. PSNR and SSIM across iterations (FL)

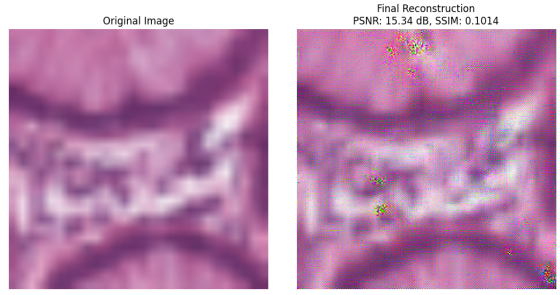


Figure 15. Original vs. reconstructed image (FL)

DLG on Baseline FL: In the absence of encryption, DLG was highly effective. Figure 13 shows the evolution of the dummy input converging visually toward the ground truth. The steady rise of PSNR and SSIM in Figure 14 confirms high-quality reconstruction. The final result in Figure 15 is nearly indistinguishable from the true input.

DLG on Selective HE: Under selective encryption, the DLG attack fails to reconstruct any meaningful input, as demonstrated in Figure 16. The PSNR and SSIM curves in Figure 17 remain extremely low throughout, and the final output (Figure 18) is indistinguishable from noise.

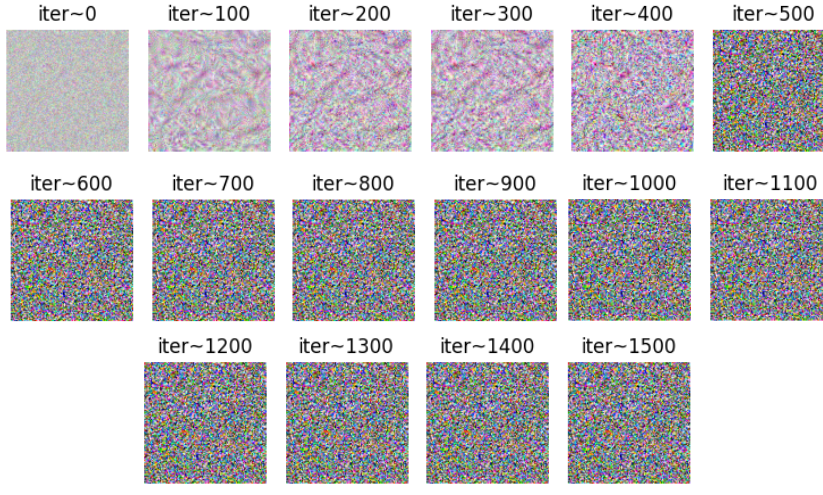


Figure 16. Image reconstruction progression (FL_SelHE)

These results provide strong empirical evidence that selective encryption, when applied to structurally sensitive model layers, can effectively neutralize gradient

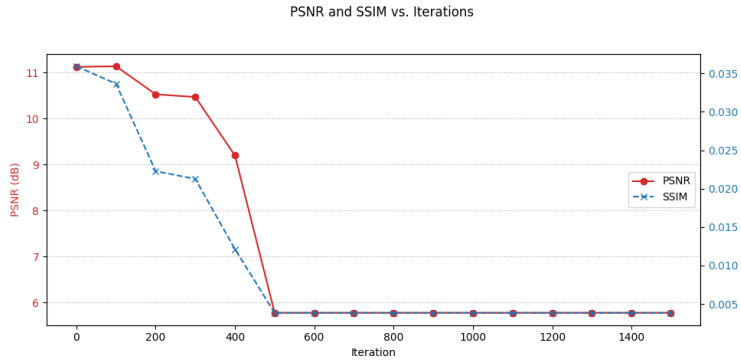


Figure 17. PSNR and SSIM across iterations (FL_SelHE)

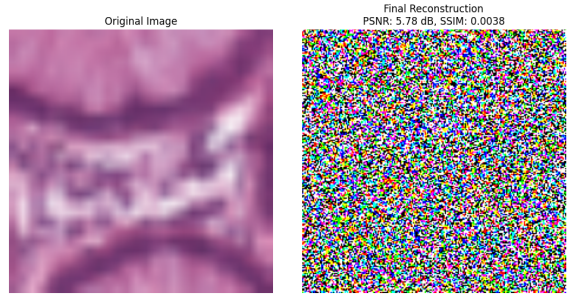


Figure 18. Original vs. reconstructed image (FL_SelHE)

based attacks without significant compromise in performance or system efficiency.

6 DISCUSSION

The findings of this study converge to form a scientifically coherent and practically relevant approach for the integration of homomorphic encryption within federated learning, particularly for privacy-sensitive applications in medical imaging. This discussion integrates empirical findings and examines their implications with an emphasis on accuracy, efficiency, security, and systemic design trade-offs. The experimental process began with a thorough assessment of the deep learning models in a centralized setting to establish a solid performance baseline. EfficientNet-B0 was identified as the superior architecture, demonstrating higher classification accuracy compared to CNNMed, MobileNetV3, and ResNet-18, while also preserving an advantageous computational efficiency. This result is in line with EfficientNet's compound scaling principle, which carefully balances the depth, width, and resolution to obtain the best mix of speed and accuracy. Furthermore, while our results focus on selective HE, we acknowledge the potential of SMPC-based approaches. In

future iterations, a hybrid configuration using SMPC for the non-encrypted layers and HE for the sensitive ones could offer an even stronger defense against multi-party collusion, though at the cost of higher communication overhead. Once we moved into the federated learning setup, we tested the behavior of the model under both IID and Non-IID data conditions. In the IID case, the model's accuracy and loss remained almost the same as those in the centralized setup. This close match shows that FedAvg can work well when the data are fairly similar across clients. Even as we increased the number of clients from five to ten, the training remained stable and consistent, suggesting that splitting the training across clients does not cause problems as long as the data are not too different. In contrast, the Non-IID setting, modelled via a Dirichlet distribution with a concentration parameter $\alpha = 0.5$, introduced a degree of statistical skew that represented real-world decentralized environments. This heterogeneity resulted in measurable, albeit moderate, reductions in performance metrics and slower convergence. These effects were due to client drift and unbalanced gradient contributions during aggregation. However, the model demonstrated consistent learning behavior, demonstrating the resilience of federated training even under adverse data distributions. The findings emphasize the importance of understanding client-level variability and highlight the potential utility of advanced aggregation techniques such as FedProx or FedNova when statistical heterogeneity becomes more pronounced. The integration of homomorphic encryption into a federated workflow is the primary contribution of this work. Empirical evaluation across three configurations unencrypted, fully encrypted, and selectively encrypted federated learning showed that full encryption resulted in only a marginal decrease in model accuracy (from 95.29 % to 95.13 %). This shows that CKKS can handle high precision deep learning tasks, as long as the encryption parameters are properly tuned. The chosen parameter set, with a polynomial modulus degree of 16 384 and a global scale of 2^{40}, effectively reduced quantisation noise and enabled multi-step encrypted computation. While full encryption offers strong privacy protection, it comes at a noticeable computational cost, especially on the client side. Training rounds were almost 48 % slower than the baseline, mainly because of the extra time required for encryption, decryption, and handling of the ciphertext. In practical deployments, particularly in edge settings with limited computational resources, this overhead may render full encryption infeasible. To mitigate this, a selective encryption strategy was devised that targets only the most privacy-sensitive layers, identified through gradient sensitivity and architectural depth. This approach preserved most of the privacy gains of full encryption while reducing client-side overhead by approximately 22 % and total round duration by nearly 30 %, relative to full encryption. These savings are nontrivial and suggest that partial encryption, when applied judiciously, constitutes a viable path toward privacy-preserving machine learning at scale. The framework's security claims were further substantiated by simulating a Deep Leakage from Gradients (DLG) attack. In the baseline federated learning setting, DLG successfully reconstructed the original training images with high perceptual quality, as reflected in the elevated PSNR and SSIM values and near-perfect visual alignment. These results

highlight that even partial gradient exposure poses serious privacy risks. However, when using selective homomorphic encryption focused on the sensitive layers, the same attack cannot recover anything meaningful from the encrypted gradients. The reconstructed images looked like random Gaussian noise, with PSNR dropping to approximately 5 dB and SSIM approaching zero, indicating that the images were completely degraded. This shows that encrypting only the most critical parts of the gradients is sufficient to stop even advanced gradient inversion attacks, while still keeping the approach practical.

7 CONCLUSION

In this research, we introduced FL_SelHE to bridge the gap between high-level privacy and system efficiency in federated learning. By applying the CKKS scheme only to layers identified through our sensitivity profiling, we successfully neutralized deep leakage attacks while maintaining the performance levels needed for complex medical architectures. Our findings suggest that protecting as little as 15 % of the model is enough to render reconstructed data useless (PSNR $\approx$ 5 dB), effectively stopping ruthless reconstruction attempts. In summary, this study shows that homomorphic encryption, especially when applied selectively to the parts most at risk, provides federated learning with a solid, practical option for privacy-focused settings. It strikes a good balance between keeping data safe, running efficiently, and fitting into current systems, thus making it a strong candidate for real-world deployment. Moving forward, we aim to explore dynamic layer selection and the combination of selective HE with differential privacy to create a truly multi-layered, adaptive security framework for even more complex data silos. Finally, testing real-world datasets from privacy-sensitive areas would help us better understand how ready the framework is for deployment and compliance in real-use cases.

REFERENCES

[1] Kairouz, P.—McMahan, H. B.: Advances and Open Problems in Federated Learning. Foundations and Trends® in Machine Learning, Vol. 14, 2021, No. 1-2, pp. 1–210, doi: 10.1561/2200000083.

[2] Khalid, N.—Qayyum, A.—Bilal, M.—Al-Fuqaha, A.—Qadir, J.: Privacy-Preserving Artificial Intelligence in Healthcare: Techniques and Applications. Computers in Biology and Medicine, Vol. 158, 2023, Art. No. 106848, doi: 10.1016/j.compbiomed.2023.106848.

[3] Maguluri, K. K.—Ganti, V. K. A. T.—Polineni, T. N. S.—Abhireddy, N.: Advancing Patient Privacy in the Era of Artificial Intelligence: A Deep Learning Approach to Ensuring Compliance with HIPAA and Addressing Ethical Challenges in Healthcare Data Security. International Journal of Medical Toxicology & Legal Medicine, Vol. 27, 2024, No. 5, pp. 217–228, https://ijmtlm.org/index.php/journal/article/view/962.

[4] Price, W. N.—Gerke, S.—Cohen, I. G.: Potential Liability for Physicians Using Artificial Intelligence. JAMA, Vol. 322, 2019, No. 18, pp. 1765–1766, doi: 10.1001/jama.2019.15064.

[5] McMahan, B.—Moore, E.—Ramage, D.—Hampson, S.—Aguera y Arcas, B.: Communication-Efficient Learning of Deep Networks from Decentralized Data. In: Singh, A., Zhu, J. (Eds.): Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research (PMLR), Vol. 54, 2017, pp. 1273–1282, doi: 10.48550/arXiv.1602.05629.

[6] Zhu, L.—Liu, Z.—Han, S.: Deep Leakage from Gradients. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (Eds.): Advances in Neural Information Processing Systems 32 (NeurIPS 2019). Curran Associates, Inc., Vol. 32, 2019, pp. 14774–14784, doi: 10.48550/arXiv.1906.08935.

[7] Dwork, C.—Roth, A.: The Algorithmic Foundations of Differential Privacy. Foundations and Trends® in Theoretical Computer Science, Vol. 9, 2014, No. 3-4, pp. 211–487, doi: 10.1561/0400000042.

[8] Evans, D.—Kolesnikov, V.—Rosulek, M.: A Pragmatic Introduction to Secure Multi-Party Computation. Foundations and Trends® in Privacy and Security, Vol. 2, 2018, No. 2-3, pp. 70–246, doi: 10.1561/3300000019.

[9] Gentry, C.—Sahai, A.—Waters, B.: Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based. In: Canetti, R., Garay, J. A. (Eds.): Advances in Cryptology – CRYPTO 2013. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 8042, 2013, pp. 75–92, doi: 10.1007/978-3-642-40041-4_5.

[10] Phong, L. T.—Aono, Y.—Hayashi, T.—Wang, L.—Moriai, S.: Privacy-Preserving Deep Learning via Additively Homomorphic Encryption. IEEE Transactions on Information Forensics and Security, Vol. 13, 2018, No. 5, pp. 1333–1345, doi: 10.1109/TIFS.2017.2787987.

[11] Wei, K.—Li, J.—Ding, M.—Ma, C.—Yang, H. H.—Farokhi, F.—Jin, S.—Quek, T. Q. S.—Poor, H. V.: Federated Learning with Differential Privacy: Algorithms and Performance Analysis. IEEE Transactions on Information Forensics and Security, Vol. 15, 2020, pp. 3454–3469, doi: 10.1109/TIFS.2020.2988575.

[12] Wang, Z.—Yu, X.—Huang, Q.—Gong, Y.: An Adaptive Differential Privacy Method Based on Federated Learning. 2024, doi: 10.48550/arXiv.2408.08909 (arXiv Preprint arXiv:2408.08909).

[13] Truex, S.—Liu, L.—Chow, K. H.—Gursoy, M. E.—Wei, W.: LDP-Fed: Federated Learning with Local Differential Privacy. Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking (EdgeSys '20), 2020, pp. 61–66, doi: 10.1145/3378679.3394533.

[14] Wu, F.—Gao, M.: Federated Learning Algorithm Based on Gaussian Local Differential Noise. In: Zeng, N., Pachori, R. B. (Eds.): Proceedings of 2024 International Conference on Machine Learning and Intelligent Computing. Proceedings of Machine Learning Research (PMLR), Vol. 245, 2024, pp. 325–339, `https://proceedings.mlr.press/v245/fan24a.html`.

[15] Zhu, H.—Li, Z.—Cheah, M.—Goh, R. S. M.: Privacy-Preserving Weighted

Federated Learning Within Oracle-Aided MPC Framework. 2020, doi: 10.48550/arXiv.2003.07630 (arXiv Preprint arXiv:2003.07630).

[16] Li, Y.—Zhou, Y.—Jolfaei, A.—Yu, D.—Xu, G.—Zheng, X.: Privacy-Preserving Federated Learning Framework Based on Chained Secure Multiparty Computing. IEEE Internet of Things Journal, Vol. 8, 2021, No. 8, pp. 6178–6186, doi: 10.1109/JIOT.2020.3022911.

[17] Kanagavelu, R.—Wei, Q.—Li, Z.—Zhang, H.—Samsudin, J.—Yang, Y.—Goh, R. S. M.—Wang, S.: CE-Fed: Communication Efficient Multi-Party Computation Enabled Federated Learning. Array, Vol. 15, 2022, Art. No. 100207, doi: 10.1016/j.array.2022.100207.

[18] Chen, L.—Xiao, D.—Xiao, X.—Zhang, Y.: Secure and Efficient Federated Learning via Novel Authenticable Multi-Party Computation and Compressed Sensing. IEEE Transactions on Information Forensics and Security, Vol. 19, 2024, pp. 10141–10156, doi: 10.1109/TIFS.2024.3486611.

[19] Zhang, C.—Li, S.—Xia, J.—Wang, W.—Yan, F.—Liu, Y.: BatchCrypt: Efficient Homomorphic Encryption for Cross-Silo Federated Learning. Proceedings of the 2020 USENIX Conference on Usenix Annual Technical Conference (USENIX ATC '20), USENIX Association, 2020, pp. 493–506.

[20] Jayaram, K. R.—Verma, A.—Verma, A.—Thomas, G.—Sutcher-Shepard, C.: MYSTIKO: Cloud-Mediated, Private, Federated Gradient Descent. 2020 IEEE 13th International Conference on Cloud Computing (CLOUD), 2020, pp. 201–210, doi: 10.1109/CLOUD49709.2020.00039.

[21] Ma, J.—Naas, S. A.—Sigg, S.—Lyu, X.: Privacy-Preserving Federated Learning Based on Multi-Key Homomorphic Encryption. International Journal of Intelligent Systems, Vol. 37, 2022, No. 9, pp. 5880–5901, doi: 10.1002/int.22818.

[22] Geyer, R. C.—Klein, T.—Nabi, M.: Differentially Private Federated Learning: A Client Level Perspective. 2017, doi: 10.48550/arXiv.1712.07557 (arXiv Preprint arXiv:1712.07557).

[23] Bonawitz, K.—Ivanov, V.—Kreuter, B.—Marcedone, A.—McMahan, H. B.—Patel, S.—Ramage, D.—Segal, A.—Seth, K.: Practical Secure Aggregation for Privacy-Preserving Machine Learning. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17), 2017, pp. 1175–1191, doi: 10.1145/3133956.3133982.

[24] Buyukates, B.—So, J.—Mahdavifar, H.—Avestimehr, S.: LightVeriFL: A Lightweight and Verifiable Secure Aggregation for Federated Learning. IEEE Journal on Selected Areas in Information Theory, Vol. 5, 2024, pp. 285–301, doi: 10.1109/JSAIT.2024.3391849.

[25] Chen, H.—Dai, W.—Kim, M.—Song, Y.: Efficient Multi-Key Homomorphic Encryption with Packed Ciphertexts with Application to Oblivious Neural Network Inference. Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS '19), 2019, pp. 395–412, doi: 10.1145/3319535.3363207.

[26] Cheon, J. H.—Kim, A.—Kim, M.—Song, Y.: Homomorphic Encryption for Arithmetic of Approximate Numbers. In: Takagi, T., Peyrin, T. (Eds.): Advances in Cryptology – ASIACRYPT 2017. Springer, Cham, Lecture Notes in Computer

Science, Vol. 10624, 2017, pp. 409–437, doi: 10.1007/978-3-319-70694-8_15.

[27] Bisong, E.: Google Colaboratory. Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners, Apress, Berkeley, CA, 2019, pp. 59–64, doi: 10.1007/978-1-4842-4470-8_7.

[28] Beutel, D. J.—Topal, T.—Mathur, A.—Qiu, X.—Fernandez-Marques, J. et al.: Flower: A Friendly Federated Learning Research Framework. 2020, doi: 10.48550/arXiv.2007.14390 (arXiv Preprint arXiv:2007.14390).

[29] Benaissa, A.—Retiat, B.—Cebere, B.—Belfedhal, A. E.: TenSEAL: A Library for Encrypted Tensor Operations Using Homomorphic Encryption. 2021, doi: 10.48550/arXiv.2104.03152 (arXiv Preprint arXiv:2104.03152).

[30] Yang, J.—Shi, R.—Wei, D.—Liu, Z.—Zhao, L.—Ke, B.—Pfister, H.—Ni, B.: MedMNIST V2 – A Large-Scale Lightweight Benchmark for 2D and 3D Biomedical Image Classification. Scientific Data, Vol. 10, 2023, No. 1, Art. No. 41, doi: 10.1038/s41597-022-01721-8.

[31] Sokolova, M.—Lapalme, G.: A Systematic Analysis of Performance Measures for Classification Tasks. Information Processing & Management, Vol. 45, 2009, No. 4, pp. 427–437, doi: 10.1016/j.ipm.2009.03.002.

[32] Davis, J.—Goadrich, M.: The Relationship Between Precision-Recall and ROC Curves. Proceedings of the 23rd International Conference on Machine Learning (ICML '06), 2006, pp. 233–240, doi: 10.1145/1143844.1143874.

[33] Powers, D. M. W.: Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation. 2020, doi: 10.48550/arXiv.2010.16061 (arXiv Preprint arXiv:2010.16061).

[34] van Rijsbergen, C. J.: Information Retrieval. Butterworth-Heinemann, United States, 1979.

[35] He, K.—Zhang, X.—Ren, S.—Sun, J.: Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[36] Howard, A.—Sandler, M.—Chen, B.—Wang, W.—Chen, L. C.—Tan, M.—Chu, G.—Vasudevan, V.—Zhu, Y.—Pang, R.—Adam, H.—Le, Q.: Searching for MobileNetV3. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 1314–1324, doi: 10.1109/ICCV.2019.00140.

[37] Tan, M.—Le, Q.: EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In: Chaudhuri, K., Salakhutdinov, R. (Eds.): Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research (PMLR), Vol. 97, 2019, pp. 6105–6114, doi: 10.48550/arXiv.1905.11946.

[38] Zhao, B.—Mopuri, K. R.—Bilen, H.: iDLG: Improved Deep Leakage from Gradients. 2020, doi: 10.48550/arXiv.2001.02610 (arXiv Preprint arXiv:2001.02610).



Mohammed El Aymene Beyat earned his Engineering degree in network administration and security from the Kasdi Merbah University, Ouargla, Algeria, in 2022. He is currently Ph.D. researcher in computer science at the same university. His interests encompass cybersecurity, artificial intelligence, advanced machine learning techniques, data privacy, and the Internet of Things (IoT).



Mohammed Kamel Benkaddour received his Engineering degree in telecommunications and his Ph.D. degrees in electronics from the University of Djillali Liabes, Sidi Bel Abbes, Algeria, in 2008 and 2020, respectively. He is currently Associate Professor with the Department of Computer Science in University of Kasdi Merbah Ouargla, Algeria. His research interests include cyber security, artificial intelligence, machine learning and data privacy.



Abdessalam Moualdi received his Engineering degree in electronics of embedded systems from Kasdi Merbah University, Ouargla, Algeria, in 2023. He is currently pursuing his Ph.D. degree in electronics at the same university. His research interests include deep multi-view clustering, artificial intelligence, advanced machine learning, data privacy and optimization.



Abdelalim Belmesmar obtained his Master's degree in security and network management from the Faculty of Computer Science and Information Technology, University of Kasdi Merbah, Ouargla, Algeria, in 2025. His research interests span the fields of cybersecurity, network protection, artificial intelligence, machine learning, information privacy, and system administration.



Mohammed El Amine Beyat received his Engineering degree in network administration and security from the Kasdi Merbah University, Ouargla, Algeria, in 2022. He is currently Ph.D. student in computer science at the same university. His research focuses on network and information security, with a particular interest in enhancing the reliability and resilience of vehicular ad hoc networks.