



RESEARCH ARTICLE

OPEN ACCESS

ADAPTIVE SENSITIVE-LAYERS BASED HOMOMORPHIC ENCRYPTION FOR SCALABLE AND SECURE FEDERATED LEARNING

Mohammed El Amine Beyat^{*1}, Mohammed Kamel Benkaddour², Ahmed Korichi³,
Mohammed El Aymene Beyat⁴, Abdessalam Moualdi⁵

^{1, 2, 3, 4}Department of Computer Science and Information Technology, Laboratory of Artificial Intelligence and Information Technologies, University of Kasdi Merbah, 30000, Ouargla, Algeria.

⁵Department of Electronics and Telecommunications, Laboratory of Electrical Engineering, University of Kasdi Merbah, 30000, Ouargla, Algeria.

¹<http://orcid.org/0009-0009-2323-1251>, ²<http://orcid.org/0000-0003-1895-7540>, ³<http://orcid.org/0000-0002-5741-4963>,

⁴<http://orcid.org/0009-0001-5275-4627>, ⁵<http://orcid.org/0009-0000-3758-8782>

E-mail: *beyat.mohammedelamine@univ-ouargla.dz, benkaddour.kamel@univ-ouargla.dz, ahmed.korichi@univ-ouargla.dz,
beyat.mohammedelaimene@univ-ouargla.dz, moualdi.abdessalam@univ-ouargla.dz

ARTICLE INFO

Article History

Received: February 19, 2026

Reviewed: March 19, 2026

Accepted: May 15, 2026

Published: September 30, 2026

Keywords:

Federated Learning,
Privacy Preservation,
Homomorphic Encryption,
Selective Encryption,
CKKS Scheme,
Sensitive Layer Protection.

ABSTRACT

Federated Learning (FL) enables collaborative model training across decentralized clients while maintaining data locality; however, the protection of exchanged model updates remains a critical security challenge. While Full Homomorphic Encryption (FHE) ensures high confidentiality, its prohibitive computational and communication overhead often renders it impractical for resource-constrained FL systems. To address this bottleneck, we propose a Selective Homomorphic Encryption (SHE) framework that optimizes the utility-efficiency trade-off. Our approach leverages a one-time gradient sensitivity analysis to identify and encrypt only the most critical model layers using the CKKS scheme, while maintaining less sensitive layers in plaintext. Experimental evaluations conducted on the BloodMNIST dataset demonstrate that this selective strategy maintains a model utility nearly identical to standard FL. The proposed framework achieved an accuracy of 96.18% with a loss of 0.1651, performing comparably to the baseline FL (96.32%, loss : 0.1628) and outperforming the fully encrypted configuration (96.05%, loss: 0.1684). Furthermore, the selective scheme significantly improved system efficiency, reducing client computation time by 12.86% and cutting encryption-related overhead by up to 41.67%. These findings suggest that selective homomorphic encryption provides a highly practical and scalable solution for privacy-preserving FL deployments in sensitive domains such as healthcare and medical imaging.



Copyright ©2026 by authors and Galileo Institute of Technology and Education of the Amazon (ITEGAM). This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

I. INTRODUCTION

Artificial intelligence (AI) is now used in many areas such as healthcare, finance, and personalized services. These domains rely on advanced predictive models, but they also have to follow strict privacy regulations and ethical rules for data use [1] [2]. Deep learning is common in diagnostic systems, clinical decision support, and other data-driven tools. In a typical centralized setup, training still requires sending all data to a single server. This central collection raises strong privacy, regulatory, and security concerns under frameworks such as GDPR and HIPAA [3]. Federated Learning (FL) provides a decentralized way to train models without collecting raw data in one place [4]. Clients keep their data locally, train the model on their own devices, and only send model updates to a central server. This reduces direct exposure of raw data, but exchanged model updates can still leak private information. Protecting these updates is therefore a central requirement in practical FL systems. Common privacy-preserving methods in FL are Differential Privacy (DP), Secure Multiparty Computation (SMPC), and Homomorphic Encryption (HE). DP adds calibrated noise to protect individual contributions but may reduce accuracy [5]. SMPC allows joint computation without revealing inputs but often leads to high communication and computation costs for large models [6].

HE allows computation directly on encrypted data [7] and provides strong protection, but fully encrypting all model parameters typically introduces large latency and overhead [8]. To reduce these costs, we propose a gradient-aware selective encryption framework based on the CKKS (Cheon-Kim-Kim-Song) scheme. A one-time gradient sensitivity analysis identifies the most sensitive layers, which are then encrypted, while less critical layers remain in plaintext. In this way we keep strong protection where it is most needed while lowering the computational burden. The mixed encrypted and plaintext workflow fits naturally into a standard FL pipeline. We implement the system using the Flower FL framework and TenSEAL for CKKS encryption [9]. MobileNetV3 Small is used on the client side due to its efficiency and strong performance in medical imaging. We evaluate the approach on the BloodMNIST dataset [10] with 5 and 10 clients under both IID and non-IID distributions. Selective HE achieves 96.18% accuracy (loss 0.1651), which is close to the FL baseline (96.32%, loss 0.1628) and better than full HE (96.05%, loss 0.1684). It also improves efficiency, reducing client computation time by 12.86% and cutting encryption overhead by up to 41.67%. These results show that selective HE offers a practical balance between privacy, accuracy, and computation cost for real FL deployments in sensitive domains such as medical imaging. The remainder of this paper is organized as follows. Section II introduces the necessary background on federated learning and encryption-based privacy preservation. Section III outlines the proposed methodology, including the system design, encryption strategies, and gradient-aware sensitivity profiling. Section IV delineates the experimental setup, encompassing the dataset, evaluation metrics, and the presentation of experimental results for both IID and non-IID distributions. Section V provides a comprehensive discussion of system-level trade-offs and practical implications. Finally, Section VI synthesizes the research findings and outlines potential trajectories for future work.

II. BACKGROUND AND RELATED WORK

Privacy-preserving Federated Learning (FL) has emerged as a cornerstone for collaborative model training without the necessity of centralizing raw datasets, a paradigm particularly vital in high-stakes domains such as healthcare and finance where data sensitivity is governed by stringent regulations [1-4]. While FL avoids direct data sharing, exchanged model updates remain vulnerable to inference attacks, necessitating additional protection mechanisms such as Differential Privacy (DP), Secure Multi-party Computation (SMPC), and Homomorphic Encryption (HE). DP protects privacy by injecting calibrated statistical noise, as demonstrated by Xia et al. [11] in their global DP FL framework; however, the resulting noise-utility trade-off often degrades the accuracy of complex deep learning models [5], [12]. Alternatively, SMPC protocols allow for private joint computation, with Liu et al. [13] exploring reduced communication overhead by decoupling computation from cryptography, yet these methods often struggle to scale with large models and high client counts [6]. In contrast, HE allows for mathematical operations directly on ciphertexts, with notable implementations including BatchCrypt by Zhang et al. [14], which optimizes CKKS batching, and MYSTIKO developed by Fang and Qian [15] using Paillier encryption. Although HE preserves model utility better than DP, the computational cost of full encryption remains a significant bottleneck [7], [8], motivating the development of Selective Homomorphic Encryption (SHE). Selective encryption protects only the most sensitive model components a strategy that harmonizes effectively with HE schemes like CKKS to reduce resource consumption while maintaining robust privacy guarantees [16], [17]. In this context, our work adopts a selective CKKS-based encryption strategy that identifies high-sensitivity layers through gradient profiling. On the BloodMNIST dataset [10], this approach achieved an accuracy of 96.18% (loss: 0.1651), performing comparably to baseline FL (96.32%) and outperforming full HE (96.05%). Crucially, our framework reduced encryption overhead by up to 41.67% and decreased total client computation time by 12.86%, demonstrating that selective HE is a highly practical solution for privacy-critical FL applications.

III. PROPOSED METHODOLOGY

III.1 SYSTEM OVERVIEW

We build a gradient-aware selective homomorphic encryption framework to provide strong privacy protection while keeping the computational cost reasonable in federated learning (FL), as illustrated in the system overview in Figure 2. The main goal is to protect the client-side model updates during transmission and aggregation, while maintaining practical usability in sensitive domains such as healthcare. We assume a semi-honest central server that follows the protocol but is not trusted with access to the actual content of client updates. To preserve confidentiality, each client encrypts selected parts of its model parameters using homomorphic encryption (HE) before transmission. This enables the server to perform mathematical operations directly on encrypted tensors such that the decrypted results correspond to operations performed on plaintext data (Figure 1). The server aggregates encrypted updates without accessing their underlying values. In the proposed selective encryption framework, only sensitive layers are encrypted, while non-sensitive parameters may remain in plaintext to reduce computational overhead. The process consists of the following steps:

1. The server initializes the global model and distributes it to all clients.
2. Each client trains the model locally on its private dataset.
3. Based on the encryption mode, the client encrypts either all layers or a subset of layers using CKKS.
4. The encrypted updates, along with any non-sensitive plaintext parameters, are transmitted to the server.
5. The server performs homomorphic aggregation on encrypted components and standard aggregation on plaintext components without accessing sensitive information.
6. Clients decrypt the aggregated encrypted components and proceed to the next training round.

We consider two encryption modes:

- Full Encryption: all model layers are encrypted.
- Selective Encryption: only layers identified as sensitive through gradient sensitivity profiling are encrypted.

Selective encryption significantly reduces computation and communication overhead while preserving model performance. In our BloodMNIST experiments, this approach achieved 96.18% accuracy and reduced client computation time by 12.86% compared to full encryption, while remaining close to the baseline FL accuracy of 96.32%.

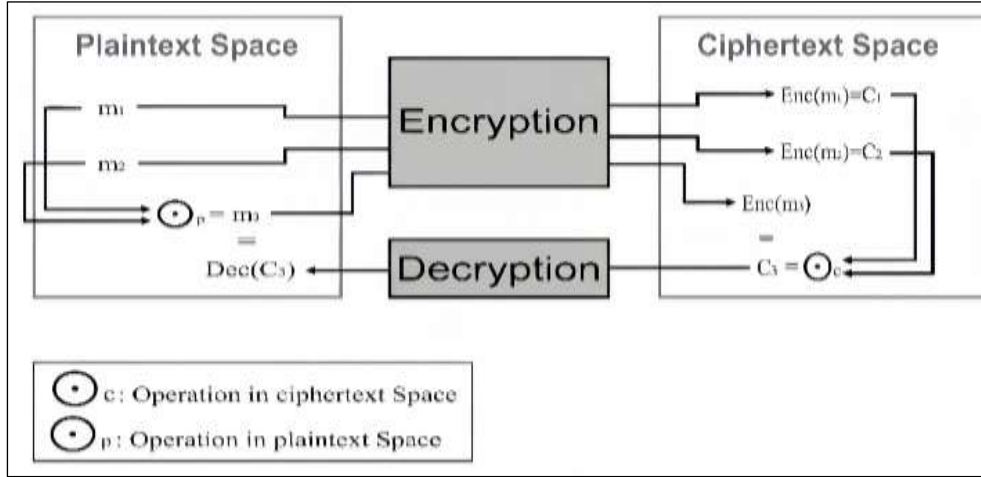


Figure 1: Schematic of the homomorphic property and its operational mapping between plaintext and ciphertext spaces.
Source: Authors, (2026).

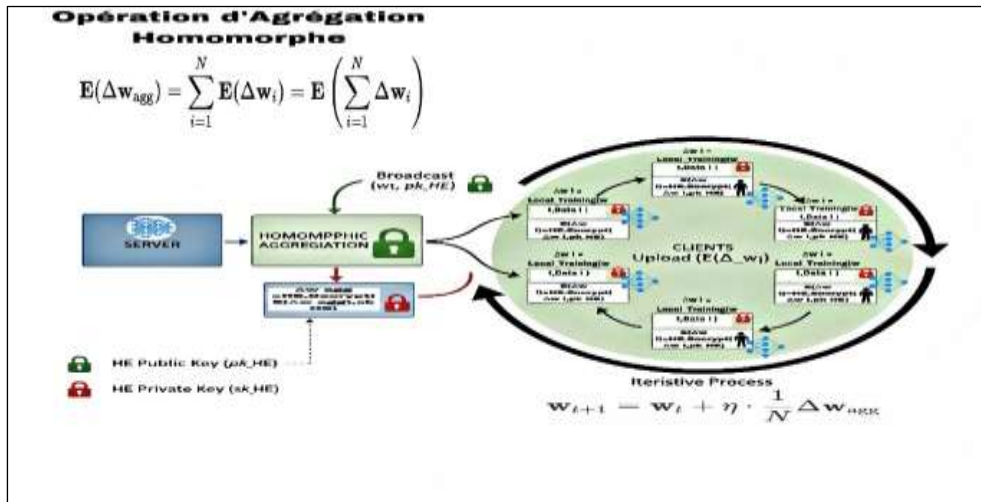


Figure 2: Architectural overview of the proposed privacy-preserving federated learning framework utilizing homomorphic encryption for secure model aggregation.
Source: Authors, (2026).

III.2 HOMOMORPHIC ENCRYPTION STRATEGY

The proposed framework implements two distinct encryption modalities: Full Homomorphic Encryption (FHE) and Selective Homomorphic Encryption (SHE).

- FHE: Encrypts the entirety of the model's parameter space, providing the highest level of data confidentiality.
- SHE: Targets only the layers identified as high-sensitivity, maintaining robust privacy while significantly alleviating the computational burden.

Both strategies leverage the CKKS (Cheon-Kim-Kim-Song) scheme [7], a leveled homomorphic encryption method grounded in the Ring Learning with Errors (RLWE) assumption. CKKS is uniquely suited for deep learning applications as it supports approximate arithmetic on encrypted vectors, accommodating the real-valued tensors used in neural networks and their inherent tolerance for minor numerical approximations. In this implementation, the standard CKKS pipeline encompassing key generation, encoding, encryption, homomorphic operations (addition and multiplication), noise management, and decryption is managed via the TenSEAL library. While FHE and SHE utilize identical underlying CKKS operations, they differ strictly in the volume of layer-wise parameters encrypted by the client.

III.2.1 Full Homomorphic Encryption Strategy

In the Full Encryption Strategy, every model parameter is encrypted on the client side prior to transmission. While this ensures maximum confidentiality throughout the Federated Learning process, it introduces substantial latency and computational overhead. Empirical measurements of this strategy indicate a client-side computation time of 75.82 s and a total round duration of 418.92 s. The systematic workflow for the FHE strategy is detailed in Algorithm 1.

Algorithm 1 Federated Learning with Full Homomorphic Encryption

Require: Global rounds T , clients K , public key P_k , secret key S_k

Ensure: Final global model W_T

```

1  Initialize global model  $W_0$ 
2   $t \leftarrow 1$ 
3  while  $t \leq T$  do
4      Select clients  $S_t$ 
5      Each client trains locally
6      Encrypt local model parameters using  $P_k$ 
7      Send encrypted updates to server
8      Server runs EncryptedFedAvg on ciphertexts
9      Broadcast aggregated encrypted model to clients
10      $t \leftarrow t + 1$ 
11 end while

```

III.3 GRADIENT SENSITIVITY PROFILING

The efficiency of the Selective Homomorphic Encryption (SHE) framework relies on identifying the specific model layers that are most critical to secure. To systematically determine these layers, we employ a gradient sensitivity profiling methodology. This approach involves computing the average gradient norm for every layer across several training batches. Layers characterized by the highest gradient magnitudes which are frequently the deeper architectural layers and the final classifier head are categorized as sensitive and prioritized for encryption. This prioritization ensures that the most informative components of the model remain protected while significantly optimizing the total computational workload. The specific procedure for this selection is detailed in Algorithm 2.

Algorithm 2 Determine Sensitivity

Require: Model M , dataset D , top fraction γ

Ensure: Sensitive layer indices I

```

1   $I \leftarrow \emptyset$ 
2  Compute gradient magnitudes for all layers of  $M$  using  $D$ 
3  Rank layers based on gradient magnitudes
4  Select top  $\gamma\%$  ranked layers
5  Add selected layer indices to  $I$ 

```

III.4 SELECTIVE HOMOMORPHIC ENCRYPTION STRATEGY

The Selective Homomorphic Encryption (SHE) strategy is the core contribution of this framework, designed to balance the trade-off between privacy and computational efficiency. Unlike the full encryption approach, SHE restricts the use of CKKS operations to a subset of model layers identified as high-sensitivity during the profiling phase. This significantly reduces the number of ciphertext operations required per round, lowering both the client's encryption latency and the server's aggregation time. As previously noted in our measurements, this strategy achieves a significant reduction in overhead compared to FHE, while maintaining high utility. The procedural execution of the selective framework is detailed in Algorithm 3. It uses the sensitivity indices returned by Algorithm 2.

Algorithm 3 Federated Learning with Selective Homomorphic Encryption

Require: Global rounds T , clients K , public key P_k , secret key S_k , sensitive layer indices I

Ensure: Final global model W_T

```
1 Initialize global model  $W_0$ 
2 for  $t = 1$  to  $T$  do
3     Select clients  $S_t$ 
4     for each client  $j \in S_t$  do
5         Receive current global model  $W_{t-1}$ 
6         Train locally on private data and update parameters
7         Encrypt only layers in  $I$  via CKKS
8         Keep remaining layers in plaintext
9         Send mixed encrypted/plaintext updates and sample size  $n_j$ 
           to server
10    end for
11    Server aggregates encrypted layers homomorphically
12    Server aggregates plaintext layers normally
13    Broadcast aggregated mixed model to clients
14    Each client decrypts layers in  $I$  and updates local model
15 end for
```

III.5 COMPUTATIONAL INFRASTRUCTURE AND IMPLEMENTATION ENVIRONMENT

To ensure the reproducibility and efficiency of the experiments, all evaluations were conducted within a controlled environment specifically configured to meet the significant computational and memory demands of privacy-preserving Federated Learning (FL). The integration of Homomorphic Encryption (HE), specifically the CKKS scheme, necessitates high-capacity memory and modern GPU acceleration to manage expanded encrypted tensors and polynomial-based arithmetic.

III.5.1 Hardware Configuration

The experiments were executed on a high-performance HP Z8 G4 workstation. A critical component of this setup is the 96 GB of RAM, which is essential for handling the substantial memory footprint of encrypted model updates during the server-side aggregation phase. Graphical acceleration was provided by a dual-GPU configuration to support both local training and cryptographic processing. The full hardware specifications are detailed in Table 1.

Table 1: Hardware specifications and execution platform.

Component	Specification
Workstation Model	HP Z8 G4
Processor (CPU)	Intel Xeon Silver 4108
System Memory (RAM)	96 GB
Primary GPU	NVIDIA GeForce RTX 3090 (24 GB VRAM)
Secondary GPU	NVIDIA GeForce RTX 2080 Ti (11 GB VRAM)
Operating System	Microsoft Windows 10 (64-bit)

Source: Authors, (2026).

III.5.2 Software Stack and Libraries

The implementation was developed within a Python-based ecosystem, specifically selecting library versions optimized for efficient memory management and high-throughput GPU kernels.

We utilized TenSEAL as the primary interface for the Microsoft SEAL C++ library to execute homomorphic operations on model parameters. Crucially, the Flower (flwr) framework was employed to orchestrate the decentralized training process, managing the communication between the server and participating clients. The core software components, including the specific versions utilized to ensure reproducibility, are detailed in Table 2.

Table 2: Software environment and core libraries.

Library / Tool	Version	Role in Framework
Python	3.12.8	Main development and execution environment
PyTorch	2.9.1	Model architecture and GPU-accelerated training
Flower (flwr)	2.1.0	Federated learning orchestration and communication
TenSEAL	0.3.16	CKKS homomorphic encryption operations
MedMNIST	2.2.3	Data acquisition and standardized benchmarking
NumPy	2.4.0	High-dimensional array and matrix processing
Matplotlib	3.10.8	Generation of static figures and distribution plots
Seaborn	0.13.2	Statistical data visualization and confusion matrice
CUDA Toolkit	13.1	Hardware-level parallel computing interface

Source: Authors, (2026).

IV. EXPERIMENTAL SETUP

IV.1 DATASET DESCRIPTION

To evaluate the proposed selective homomorphic encryption framework, we utilized the BloodMNIST dataset, which is a subset of the MedMNIST v2 benchmark [10]. BloodMNIST is a specialized medical imaging dataset for blood cell classification, providing a realistic and high-stakes test case for privacy-preserving Federated Learning (FL) due to the sensitivity of biomedical data. The dataset comprises 17,092 microscopy images categorized into eight distinct blood cell classes. Each sample is an RGB image with a standardized resolution of 28 × 28 pixels. For our experiments, the data was partitioned into 11,959 training images, 1,712 validation images, and 3,421 test images. The structural details of the dataset are summarized in Table 3, and representative samples of the eight cell types are visualized in Figure 3.

Table 3: Overview of the BloodMNIST dataset used in the experiments.

Aspect	Details
Modality	Blood Cell Microscopy
Total Samples	17,092
Training / Validation / Test	11,959 / 1,712 / 3,421
Image Format	RGB, 28 × 28 pixels
Classes	8 blood cell types
Task	Multi-class classification

Source: Authors, (2026).

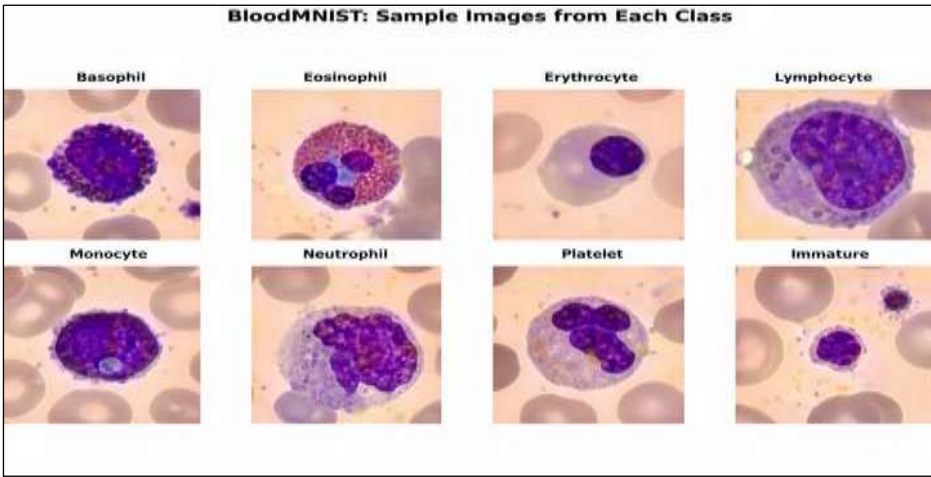


Figure 3: Example images from the eight classes of the BloodMNIST dataset.
Source: Authors, (2026).

IV.1.1 Preprocessing and Data Augmentation

To ensure stable convergence and improve the generalization capabilities of the architectures, we implemented a standardized preprocessing pipeline:

- **Tensor Conversion** : All images were converted into PyTorch tensors to enable efficient parallel processing on the GPU.
- **Normalization** : Pixel values were normalized using the ImageNet mean (μ) and standard deviation (σ). This ensures the input distribution is compatible with the pretrained backbones. Each pixel x is transformed into x_{norm} using the following equation:

$$x_{norm} = \frac{x - \mu}{\sigma} \quad (1)$$

- **Data Augmentation** : To mitigate overfitting particularly given the small 28×28 resolution we applied stochastic augmentations during training, including random horizontal and vertical flips, minor rotations (up to 15°), and color jittering.

As shown in Figure 4, the class distribution across the eight categories is maintained to ensure the model learns to identify even minority cell types effectively.

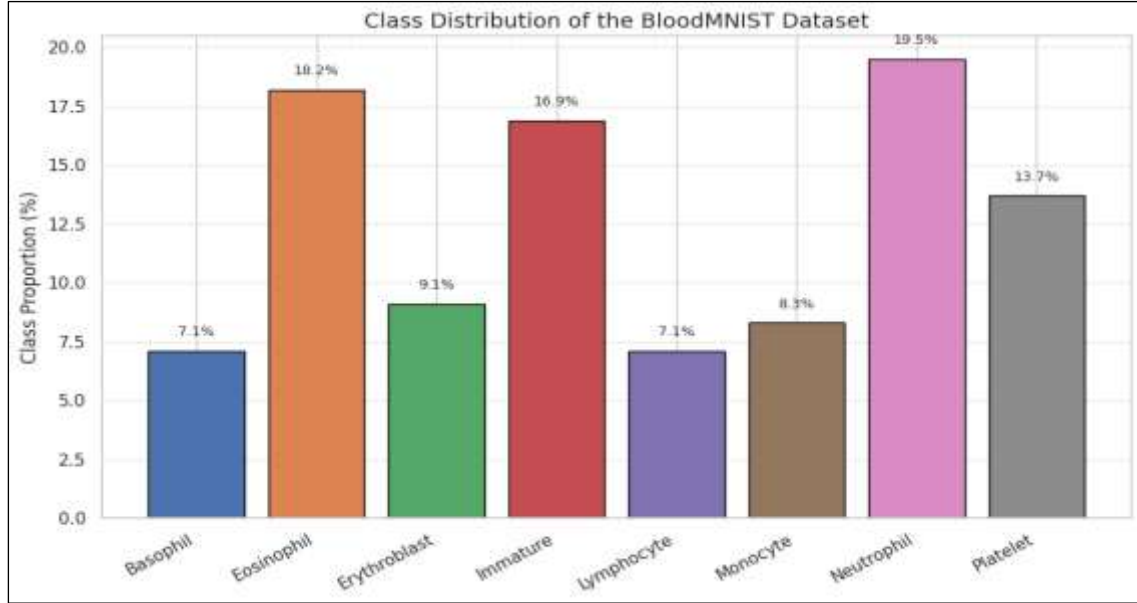


Figure 4: Class distribution of the eight BloodMNIST categories.
Source: Authors, (2026).

IV.2 PERFORMANCE EVALUATION METRICS

To provide a comprehensive assessment of the proposed framework, we benchmarked its performance across centralized, federated, and encrypted learning configurations. This multi-faceted evaluation ensures a robust understanding of the system's behavior across diverse operational environments, moving beyond the limitations of a single-scenario analysis. The performance analysis is centered on two primary, interconnected pillars:

- **Classification Utility**: Measuring the model's ability to maintain high diagnostic accuracy and reliability when transitioning from a centralized environment to a decentralized, encrypted one.
- **Computational Efficiency** : Quantifying the resource demands and latency introduced by cryptographic operations and network communication.

This dual-focus approach is essential because, in the context of privacy-preserving medical AI, classification accuracy alone does not fully characterize the practical feasibility or the deployment costs of the system. By analyzing the interplay between utility and overhead, we can identify the optimal configurations for real-world clinical integration.

IV.2.1 Classification Performance

To rigorously evaluate the predictive capabilities of our models across the eight classes of the BloodMNIST dataset, we utilized four standard quantitative metrics: Accuracy [18], Precision [19], Recall [20], and the F1-score [21]. These metrics collectively offer a holistic view of the model's diagnostic reliability and its ability to distinguish between various hematological cell types.

- **Accuracy**: The overall proportion of correctly identified samples across the entire dataset.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

- Precision: The ratio of true positive predictions to the total predicted positives, reflecting the model's exactness or ability to avoid false positives.

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

- Recall (Sensitivity): The ratio of true positive predictions to all actual positives in the class, indicating the model's completeness or ability to avoid false negatives.

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

- F1-score: The harmonic mean of Precision and Recall, providing a balanced measure that is particularly critical in scenarios involving data imbalance.

$$F1 - score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (5)$$

All metrics were derived from multi-class confusion matrices. This allows for a granular analysis of misclassification patterns, helping to identify specific cell classes that the model may frequently confuse, which is essential for medical diagnostic accuracy.

IV.2.2 Computational Overhead

Beyond classification accuracy, the integration of Homomorphic Encryption (HE) within a Federated Learning (FL) framework necessitates a rigorous evaluation of computational efficiency. To quantify the performance trade-offs associated with different cryptographic configurations, we monitored the following critical overhead metrics:

- Training Time: The duration required for each client node to perform local gradient computation on its respective data partition.
- Encryption Time : The latency introduced by the CKKS scheme during the transformation of plaintext model updates into ciphertexts.
- Decryption Time: The time required to restore aggregated ciphertext results into a plaintext format for model updating.
- Aggregation Time: The computational effort expended by the central server to perform homomorphic additions on encrypted updates versus standard averaging on plaintext parameters.
- Serialization Time: The overhead associated with encoding, packing, and preparing model parameters for transmission across the network.
- Total Round Time T_{round} : The cumulative duration of a single communication cycle, defined by the following relation :

$$T_{round} = T_{train} + T_{enc} + T_{serial} + T_{agg} + T_{dec} \quad (6)$$

These measurements allow for a multidimensional comparison between Baseline FL, Full HE, and Selective HE strategies. As demonstrated in Section IV.6, these metrics confirm that the selective encryption approach significantly mitigates computational bottlenecks while preserving the high utility required for medical imaging tasks.

IV.3 MODEL TRAINING STRATEGY

To ensure methodological consistency across the centralized, federated, and encrypted experimental frameworks, a uniform training strategy was maintained throughout all evaluations. These parameters were meticulously selected to facilitate stable convergence and mitigate the risk of overfitting on the BloodMNIST dataset, providing a reliable baseline for comparing the different architectural and privacy configurations. The core training configuration was defined as follows:

- Batch Size: 64
- Total Epochs: 12
- Initial Learning Rate: 3×10^{-4}
- Optimizer: AdamW [22]
- Loss Function: Categorical Cross-Entropy (CCE)
- Learning Rate Scheduler: Cosine Annealing [23]

We utilized Categorical Cross-Entropy as the primary objective function, complemented by a Cosine Annealing learning rate schedule to ensure a smooth decay of the learning rate toward its minimum value. These choices represent best practices in modern deep learning pipelines and are particularly effective for the complex feature spaces often encountered in medical imaging tasks.

IV.3.1 Categorical Cross-Entropy Loss

To optimize the model for multi-class classification, we employed Categorical Cross-Entropy (CCE) as the primary training objective. This loss function is particularly effective for the BloodMNIST dataset as it quantifies the divergence between the predicted probability distribution and the actual ground-truth labels. By applying a logarithmic penalty to incorrect or low-confidence predictions, CCE incentivizes the model to maximize the probability assigned to the correct class, leading to more robust and accurate classification across the eight cell categories. For a dataset containing N samples and C classes, the total loss L is defined as follows:

$$L = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^C y_{i,j} \cdot \log(\hat{y}_{i,j}) \quad (7)$$

In this equation:

- $y_{i,j}$ represents the one-hot encoded ground truth label (where the value is 1 for the correct class and 0 otherwise).
- $\hat{y}_{i,j}$ represents the predicted probability for class j of sample i , typically generated by the model's final softmax layer.

IV.3.2 AdamW Optimizer

To optimize the model parameters, we utilized the AdamW (Adaptive Moment Estimation with Weight Decay) optimizer [24]. AdamW is a specialized variant of the Adam algorithm that improves generalization by decoupling weight decay from the adaptive gradient update. In traditional Adam, the weight decay term is often tied to the adaptive learning rate, which can lead to suboptimal regularization in deep networks. By separating these two components, AdamW ensures that the weight decay λ is applied directly to the weights w_t , resulting in a more stable training trajectory. The update rule at each step t is defined as follows :

$$w_{t+1} = w_t - \eta_t \left(\frac{m_t}{\sqrt{v_t + \epsilon}} + \lambda w_t \right) \quad (8)$$

Where m_t and v_t denote the first and second moment estimates, respectively, η_t is the learning rate, λ is the weight decay coefficient, and ϵ is a small constant for numerical stability. We used the standard configuration with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1 \times 10^{-8}$.

IV.3.3 Cosine Annealing Learning Rate Scheduling

To ensure a stable and efficient training process, we adopted a Cosine Annealing learning rate scheduler [25]. This approach dynamically adjusts the learning rate η_t following a cosine decay pattern, starting from a maximum value η_{max} and gradually decreasing toward a minimum value η_{min} over the training process. The learning rate at epoch t is defined as:

$$\eta_t = \eta_{min} + \frac{1}{2} (\eta_{min} - \eta_{max}) \left(1 + \cos \left(\left(\frac{T_{cur}}{T_{max}} \right) \pi \right) \right) \quad (9)$$

Where:

- η_{max} is the initial learning rate (set to 3×10^{-4}).
- η_{min} is the target minimum learning rate (set to 0 to allow full convergence).
- T_{cur} denotes the current epoch,
- T_{max} is the total number of epochs (12).

This scheduling strategy avoids abrupt changes in the learning rate, enabling smoother convergence and improved optimization over complex loss landscapes such as those encountered in the BloodMNIST dataset.

IV.4 CENTRALIZED BENCHMARKING OF DEEP LEARNING ARCHITECTURES

To identify the most effective backbone for the BloodMNIST dataset, we first conducted a series of centralized benchmarking experiments. Establishing a robust centralized baseline is essential for accurately quantifying the performance trade-offs introduced later by federated distribution and homomorphic encryption. We evaluated a diverse selection of state-of-the-art convolutional and hybrid architectures known for their efficacy in image classification and their suitability for resource-constrained medical and federated environments. The following models were evaluated:

- MobileNetV3 Small [26]
- EfficientNet-B0 [27]
- EfficientFormerV2-S0 [28]
- ResNet-18 [29]
- ConvNeXt-Tiny [30]

All models were trained using the uniform hyperparameter configuration detailed in Section IV.3: a training duration of 12 epochs, a batch size of 64, an initial learning rate of 3×10^{-4} , the AdamW optimizer, and a cosine annealing learning rate schedule. This consistency ensures that the performance variations observed are attributable to the architectural designs rather than discrepancies in the training protocol.

IV.4.1 Architectural Configurations for Performance Benchmarking

To identify the most effective backbone for a privacy-preserving federated framework, we evaluated a diverse selection of deep learning architectures, encompassing both traditional Convolutional Neural Networks (CNNs) and hybrid CNN-transformer models. These architectures were chosen to represent a broad spectrum of accuracy and computational efficiency trade-offs. Convolutional Neural Networks (CNNs).

- MobileNetV3 Small: Designed for resource-constrained environments, this model leverages depthwise separable convolutions and squeeze-and-excitation (SE) blocks. Its lightweight footprint makes it particularly suitable for the communication-sensitive nature of federated settings.
- EfficientNet-B0 : This architecture utilizes a compound scaling method that jointly optimizes depth, width, and resolution. It provides a strong representational capacity while maintaining a relatively low parameter count.
- ResNet-18 : A classical baseline in medical imaging, ResNet-18 incorporates residual connections (skip connections) to mitigate the vanishing gradient problem, ensuring stable training and robust feature extraction.

Hybrid and Modern Architectures

- EfficientFormerV2-S0 : This hybrid model integrates standard convolutions with lightweight attention mechanisms. It aims to capture both local spatial features and global dependencies, mimicking transformer performance without the heavy overhead.
- ConvNeXt-Tiny : This model represents a modernized CNN design, incorporating larger kernels, LayerNorm, and specialized activation functions inspired by the Swin Transformer. It bridges the gap between pure CNNs and vision transformers while maintaining a purely convolutional structure.

Collectively, these five architectures provide a comprehensive benchmark, allowing us to analyze how different structural designs handle the trade-offs between local training speed, communication overhead, and classification performance on the BloodMNIST dataset.

IV.4.2 Centralized Model Performance

The performance of the candidate architectures was evaluated on the BloodMNIST test set to determine the most suitable backbone for our decentralized framework. Table 4 provides a comprehensive breakdown of the precision, recall, F1-score, and accuracy for each model over 12 training epochs.

Table 4 : Performance Comparison of Centralized Deep Learning Models on BloodMNIST.

Model	Ep.	Prec. (%)	Rec. (%)	F1 (%)	Acc. (%)
MobileNetV3 Small	12	96.56	96.54	96.55	96.64
EfficientNet-B0	12	96.07	96.09	96.08	96.17
EfficientFormerV2-S0	12	96.02	96.03	96.02	96.08
ResNet-18	12	95.39	95.35	95.37	95.44
ConvNeXt-Tiny	12	94.68	94.68	94.68	94.77

Source: Authors, (2026).

The graphical comparison in Figure 5 further highlights these performance disparities, showing that MobileNetV3 Small consistently outperformed the other architectures in classification accuracy. Beyond raw accuracy, this model demonstrated a superior balance of precision and recall, which is critical for medical imaging tasks where both false positives and false negatives carry high costs. Given this optimal balance between predictive power and computational efficiency, MobileNetV3 Small was selected as the base architecture for all subsequent Federated Learning (FL) and Homomorphic Encryption (HE) experiments. This choice is further supported by the confusion matrix presented in Figure 6, which confirms the model's ability to maintain high classification correctness across all eight distinct classes in the BloodMNIST dataset.

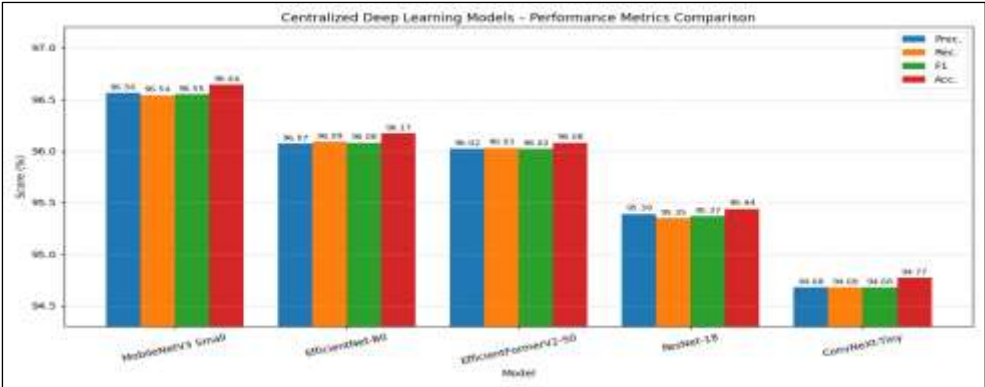


Figure 5: Performance comparison of centralized architectures on BloodMNIST.

Source: Authors, (2026).

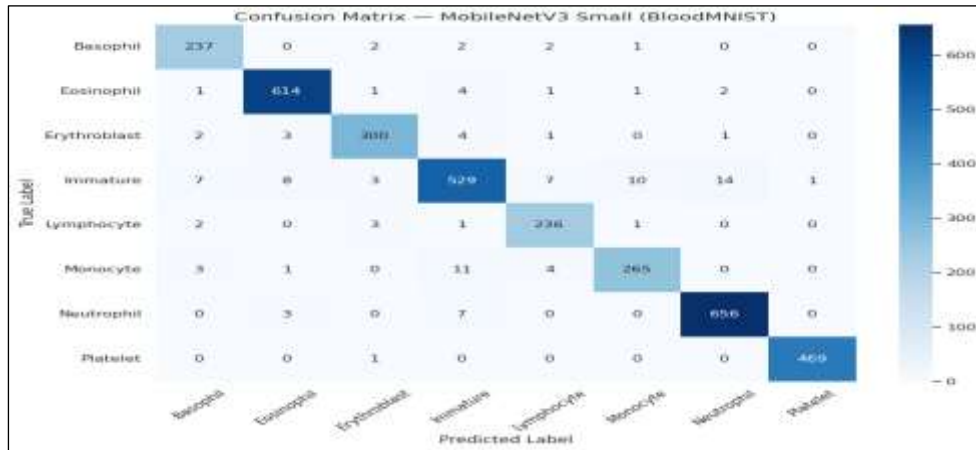


Figure 6: Confusion matrix for MobileNetV3 Small predictions on the BloodMNIST test set.
Source: Authors, (2026).

IV.5 FEDERATED LEARNING UNDER IID AND NON-IID DISTRIBUTIONS

Following the establishment of the centralized baseline, we evaluated the performance of the MobileNetV3 Small architecture within a federated learning framework. This phase of the study focuses on the model's behavior when trained on distributed data under varying levels of heterogeneity, specifically comparing Independent and Identically Distributed (IID) and Non-IID scenarios. The Non-IID partition was generated using a Dirichlet distribution with a concentration parameter $\alpha = 0.5$. All federated experiments were orchestrated using the FedAvg aggregation algorithm across varying client scales (5 and 10 clients). The specific hyperparameters and experimental configurations used to ensure a fair comparison across all tests are detailed in Table 5.

Table 5: Federated Learning Configuration.

Hyperparameter	Settings
Client Model	Client Model MobileNetV3 Small
Clients	5 and 10
Data Distribution	IID, Non-IID (Dirichlet $\alpha = 0.5$)
Rounds	10
Local Epochs	1
Batch size	64
Learning Rate	0.0003
Aggregation Method	FedAvg

Source: Authors, (2026).

IV.5.1 IID Distribution

In the Independent and Identically Distributed (IID) setting, the data was partitioned such that each client received a balanced and representative subset of the total dataset. This distribution is visualized for the 5 client case in Figure 7, where each client possesses a uniform share of the different classes within the BloodMNIST dataset. As illustrated by the training curves in Figure 8, both the 5 client and 10 client federated setups converged smoothly, exhibiting stable performance throughout the training process. The detailed numerical comparison between the centralized baseline and the federated configurations is presented in Table 6.

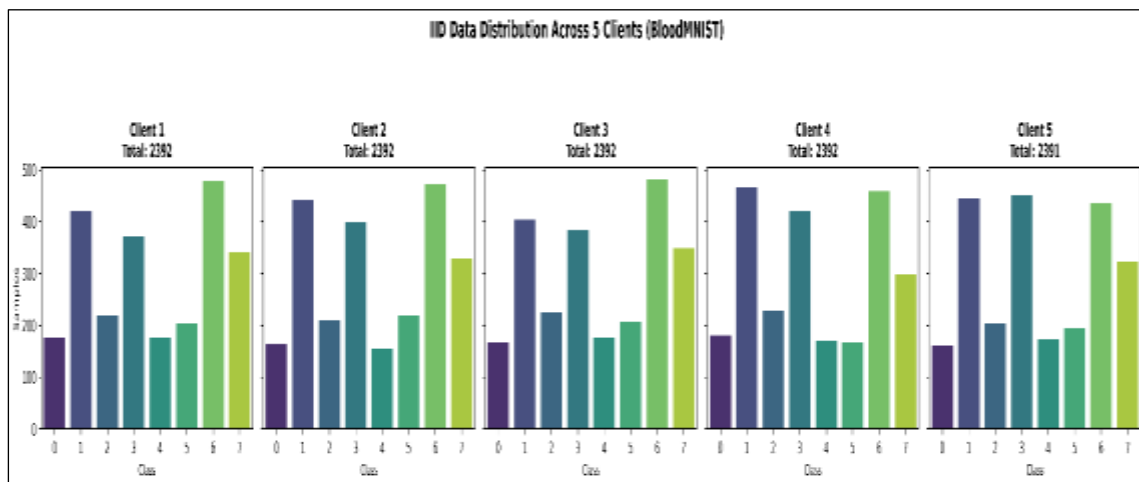


Figure 7: IID data distribution across five clients on BloodMNIST.
Source: Authors, (2026).

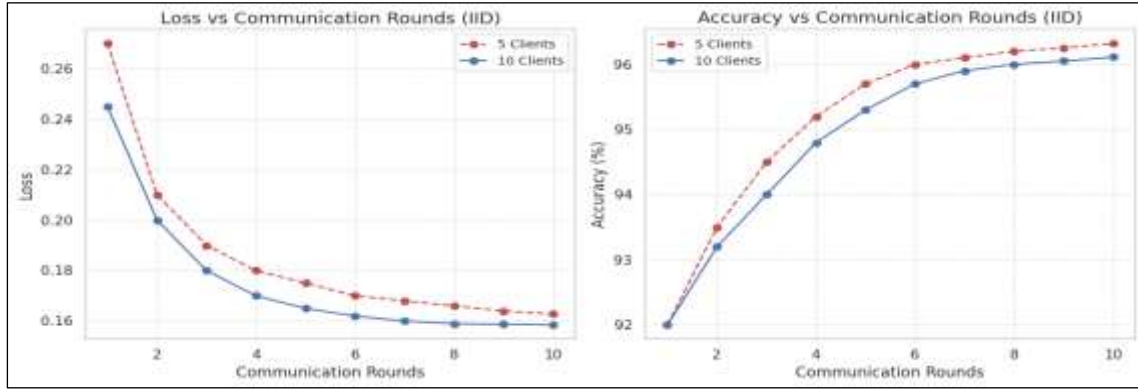


Figure 8: Federated learning accuracy and loss (IID setup).

Source: Authors, (2026).

Table 6: Centralized vs. Federated Accuracy and Loss (IID).

Method	Accuracy (%)	Loss
Centralized Model	96.64	0.1421
FL (5 Clients)	96.32	0.1628
FL (10 Clients)	96.11	0.1584

Source: Authors, (2026).

The results demonstrate that the Federated Learning configurations achieve near-parity with the centralized baseline, maintaining a high classification accuracy with a negligible utility gap of within 0.53%. This suggests that under IID conditions, the decentralized nature of FL does not significantly compromise the model's predictive power.

IV.5.2 NON-IID Distribution

To evaluate the robustness of the framework under more realistic, heterogeneous data conditions, we implemented a Non-IID setting using a Dirichlet distribution ($\alpha = 0.5$). As illustrated in Figure 9, this configuration creates significantly imbalanced client datasets, where class distributions vary across the five participating nodes. The training performance for this challenging setup is captured in Figure 10. While the accuracy and loss curves exhibit slightly more fluctuation compared to the IID case due to the gradients being calculated on divergent local data the model still achieves stable convergence. A comprehensive comparison between IID and Non-IID performance across different client scales is provided in Table 7.

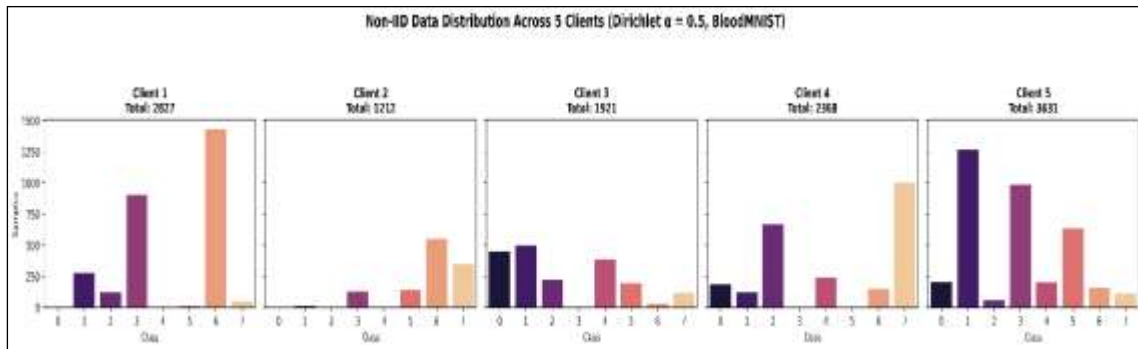


Figure 9: Non-IID data distribution across five clients.

Source: Authors, (2026).

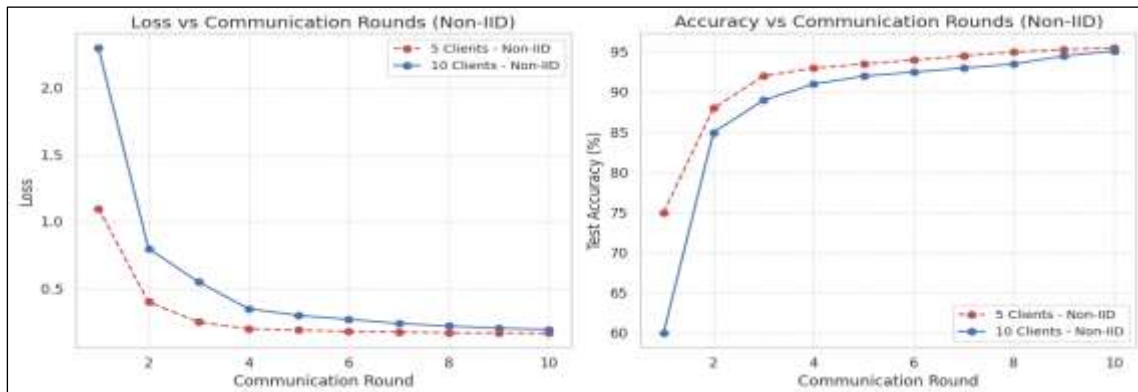


Figure 10: Federated learning accuracy and loss (Non-IID setup).

Source: Authors, (2026).

Table 7: Federated Performance: IID vs. Non-IID Comparison.

Data Type / Clients	Accuracy (%)	Loss
IID (5)	96.32	0.1628
IID (10)	96.11	0.1584
Non-IID (5)	95.50	0.1664
Non-IID (10)	95.12	0.1953

Source: Authors, (2026).

The results indicate that the Non-IID distribution introduces a minor performance penalty. Specifically, the accuracy for the 5 client setup dropped by 0.82%, while the 10 client setup saw a decrease of 0.99% compared to their IID counterparts. Despite this increased complexity and the resulting rise in loss (peaking at 0.1953 for 10 clients), the model maintains a high level of classification performance, confirming the framework's resilience to data heterogeneity.

IV.6 FEDERATED LEARNING WITH HOMOMORPHIC ENCRYPTION

To ensure robust privacy during the decentralized training process, we integrated the CKKS (Cheon-Kim-Kim-Song) homomorphic encryption scheme into our Federated Learning framework. This integration allows the central server to aggregate model updates without ever accessing the plaintext parameters. We evaluated the impact of encryption on both classification accuracy and computational overhead using the MobileNetV3 Small architecture on the BloodMNIST dataset. Both Full Encryption and Selective Encryption strategies were analyzed to determine the optimal trade-off between privacy and efficiency.

IV.6.1 Encryption Configuration

The homomorphic encryption was implemented using the following specific CKKS parameters to ensure a balance between security and the precision of the underlying floating-point arithmetic:

- Polynomial Modulus Degree (N): 16,384 providing a high security level while accommodating complex model structures.
- Coefficient Modulus Chain : [60, 40, 40, 60] tailored to manage the multiplicative depth required for the aggregation of deep learning weights.
- Global Scale : 2^{40} ensuring high precision for the encoding of fractional model parameters.
- Encoding Scheme : CKKS vectorized packing utilized to efficiently map multiple model parameters into single ciphertexts, significantly reducing communication overhead.

For the Selective Encryption mode, we employed a gradient sensitivity profiling technique to identify the most critical model components. We targeted the top 15% of layers ranked by their sensitivity scores, which predominantly included the deeper convolutional blocks and the final classifier layers. By focusing encryption on these high-sensitivity regions, we aimed to protect the most informative parts of the model while minimizing the overall computational latency.

IV.6.2 Comparative Evaluation of Model Utility and Classification Performance

We evaluated the impact of different encryption configurations on model performance using the BloodMNIST dataset. The comparative results for Baseline FL, Full HE, and Selective HE are summarized in Table 8, detailing both the accuracy and loss metrics. As shown in the training curves in Figure 11, all three configurations demonstrate stable convergence and high performance. The experimental results reveal several key insights :

- High Performance Retention : Model accuracy remains remarkably high under encryption, demonstrating that the CKKS scheme preserves the underlying mathematical structure of the updates.
- Minimal Utility Gap : Selective HE achieves an accuracy of 96.18%, staying very close to the Baseline FL accuracy of 96.32%. This confirms that encrypting only high-sensitivity layers is sufficient to maintain model utility.
- Noise Impact : Full HE exhibits a slightly higher loss (0.1684) compared to the other methods. This increase is attributed to the accumulation of ciphertext noise during the homomorphic aggregation of every model layer, though it does not significantly degrade the final classification accuracy.

Table 8: Model Accuracy and Loss under Different Encryption Configurations (BloodMNIST).

Method	Accuracy (%)	Loss
Baseline FL	96.32	0.1628
Full HE	96.05	0.1684
Selective HE	96.18	0.1651

Source: Authors, (2026).

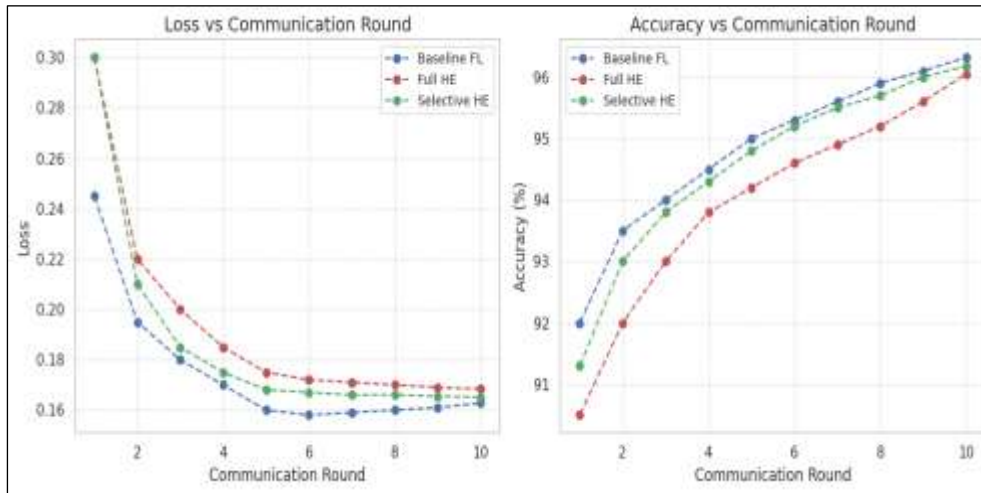


Figure 11: Training accuracy and loss for Baseline FL, Full HE, and Selective HE.
Source: Authors, (2026).

IV.6.3 Quantitative Assessment of Computational Efficiency and Resource Overhead

We evaluated the computational efficiency of our framework by measuring client and server overhead across several stages, including local training, encryption, serialization, aggregation, and total end-to-end round time. The detailed numerical comparison for the BloodMNIST dataset is summarized in Table 9. As illustrated in Figure 12, while the implementation of homomorphic encryption (HE) inherently increases the total round time compared to the baseline, the Selective HE configuration offers a significant optimization. Specifically, it achieves a 14.25% reduction in end-to-end round time compared to Full HE. The source of this efficiency is further detailed in the client-side execution breakdown shown in Figure 13. By targeting only high-sensitivity layers, Selective HE reduces the encryption overhead by 41.67% and serialization time by 42.14% compared to the Full HE baseline. Overall, these optimizations lead to a 12.86% faster client computation time, proving that the gradient-aware selection successfully mitigates the heavy computational costs of the CKKS scheme while maintaining high performance.

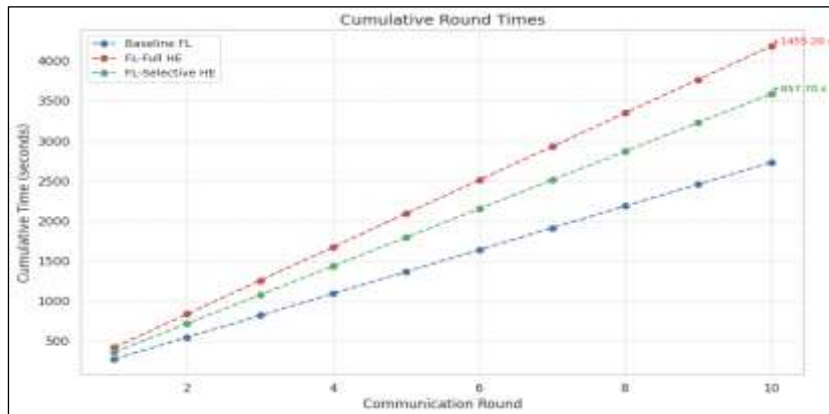


Figure 12: Total training round time for Baseline FL, Full HE, and Selective HE.
Source: Authors, (2026).



Figure 13: Client-side execution time breakdown for Baseline FL, Full HE, and Selective HE.
Source: Authors, (2026).

Table 9: Execution Time Comparison Across FL Configurations (BloodMNIST).

Computation Stage (s)	FL	Full HE	SelectiveHE	FL→Full(%)	FL→Sel (%)	Full→Sel(%)
Local Training Phase	54.12	54.28	54.20	+0.30%	+0.15%	-0.15%
Encryption Overhead	N/A	12.84	7.49	N/A	N/A	-41.67%
Decryption Overhead	N/A	1.52	0.98	N/A	N/A	-35.53%
Model Serialization	N/A	5.91	3.42	N/A	N/A	-42.14%
Model Deserialization	N/A	1.27	0.95	N/A	N/A	-25.20%
Client Computation Time	54.12	75.82	66.06	+40.07%	+22.11%	-12.86%
End-to-End Round Time	273.40	418.92	359.17	+53.22%	+31.33%	-14.25%

Source: Authors, (2026).

V. DISCUSSIONS

The results of this study suggest that adding homomorphic encryption to federated learning gives a workable balance between privacy, computation cost, and model accuracy. Here we go through the main observations from the centralized experiments, the federated runs under IID and non-IID splits, and the encrypted FL setups with full and selective CKKS-based encryption. Centralized benchmarking was first used to select a baseline model. Among all tested architectures, MobileNetV3 Small reached the best accuracy on BloodMNIST (96.64%), ahead of EfficientNet-B0, EfficientFormerV2-S0, ResNet-18, and ConvNeXt-Tiny. Its lightweight design made it a practical choice for federated and encrypted training, so we used it in all later experiments. The federated learning experiments showed stable performance in both IID and non-IID settings. In the IID case, the model reached 96.32% (5 clients) and 96.11% (10 clients), which is close to the centralized result. Under non-IID conditions (Dirichlet $\alpha = 0.5$), the accuracy dropped to 95.50% (5 clients) and 95.12% (10 clients). This drop reflects the extra difficulty caused by heterogeneous data, but the system still converged well, which matters in medical scenarios where imbalance in data is common. The integration of CKKS homomorphic encryption showed that encryption has only a small effect on accuracy. The Baseline FL reached 96.32%, Full HE reached 96.05%, and Selective HE reached 96.18%. Selective encryption, which protects only the top 15% most sensitive layers, stayed close to the baseline while encrypting far fewer parameters. The differences in efficiency were more visible. Full HE increased the round time from 273.40 s to 418.92 s (53.22% overhead). Most of this came from ciphertext generation, working with encrypted tensors during aggregation, and the cost of serialization. Selective encryption reduced these costs, bringing the round time down to 359.17 s.

This corresponds to a 31.33% overhead relative to baseline FL and about a 14.25% reduction in round time compared with Full HE, while cutting encryption-related cost by more than 40%. This gain comes from encrypting only the layers marked as most sensitive, which reduces ciphertext size and the amount of processing. Selective encryption also makes deployment easier in practice. Fully encrypted FL systems can be too heavy for many medical centers, edge devices, or low-power GPUs. Focusing protection on the most sensitive parts keeps the system in line with regulations such as GDPR and HIPAA, while avoiding the full overhead of encrypting every parameter. During implementation we also ran into several practical issues. Flower does not support encrypted tensors by default, so we had to write our own logic for serialization, communication, and aggregation of ciphertexts. This experience highlights the need for future FL frameworks to explicitly integrate native support for homomorphic encryption and secure aggregation rather than treating these mechanisms as external add-ons, in order to lower the implementation barrier and improve system usability. The gradient-aware selective encryption method successfully achieved its primary design objectives preserving model accuracy while safeguarding privacy-sensitive parameters and mitigating the computational cost observed in fully encrypted FL configurations. Importantly the experimental findings indicate that comprehensive model encryption is not a strict prerequisite for achieving strong privacy guarantees. By identifying and encrypting only the most sensitive layers, encryption can be applied in a principled and targeted manner resulting in an FL system that remains secure while being computationally efficient and practically deployable, particularly in privacy critical application domains such as medical imaging and financial analytics.

VI. CONCLUSIONS

This work presents a privacy-preserving federated learning framework that was designed with practical deployment in mind and relies on CKKS homomorphic encryption combined with a gradient-aware selective encryption strategy. Instead of encrypting the entire model, only layers identified as privacy-sensitive are protected, which allows client updates to remain confidential while avoiding unnecessary computational cost; this balance is especially important for application areas such as healthcare and finance, where both privacy and efficiency are required. The BloodMNIST dataset was used for the experimental evaluation and MobileNetV3 Small was selected as the client-side architecture due to its lightweight nature and strong baseline performance. Across both IID and non-IID data distributions, the framework maintained stable behavior, suggesting that the proposed design does not rely on overly ideal data assumptions. In the IID setting, the federated model achieved accuracies of 96.32% with 5 clients and 96.11% with 10 clients, which remain close to the centralized reference accuracy of 96.64%. When the data distribution became non-IID, accuracy decreased slightly to 95.50% and 95.12% for 5 and 10 clients respectively, yet the degradation was limited, indicating that the system remains robust even when client data differ considerably. The encryption-focused experiments made the cost of privacy more explicit, particularly in terms of training time per communication round. Applying full homomorphic encryption resulted in an accuracy of 96.05 % ; however, the average round duration increased from 273.40 s to 418.92 s, corresponding to an overhead of 53.22%, which may be restrictive in realistic federated settings. By contrast, the selective homomorphic encryption approach limited encryption to the top 15% of layers ranked by sensitivity, reducing the round time to 359.17 s, while still achieving 96.18% accuracy. This represents a 31.33% overhead compared to baseline federated learning, and an improvement of approximately 14.25% relative to full encryption, showing that selective protection can significantly reduce cost without weakening privacy guarantees.

A number of practical problems came up during implementation, especially when homomorphic encryption was integrated into the federated learning pipeline. The Flower framework does not currently provide native support for encrypted tensors, which required manual handling of ciphertext serialization, transmission, and aggregation, adding complexity to the system design. These limitations suggest that future federated learning frameworks should explicitly include secure computation primitives, including homomorphic encryption, to reduce development effort and improve reliability. The proposed gradient-aware selective CKKS encryption framework offers a realistic approach to privacy preservation in federated learning by avoiding full-model encryption while still protecting the most sensitive components. The results show that model performance can be preserved, overhead can be reduced, and privacy risks can be mitigated at the same time without relying on overly complex assumptions. Future research will examine stronger adversarial models, adaptive encryption strategies that evolve during training, as well as hybrid approaches that combine homomorphic encryption with differential privacy or secure aggregation. In addition, evaluation on larger and more diverse medical datasets will be necessary to better understand scalability and suitability for real-world deployment.

VII. AUTHOR'S CONTRIBUTION

Conceptualization: Mohammed El Amine Beyat, Mohammed El Aymene Beyat, Mohammed Kamel Benkaddour, And Ahmed Korichi.

Methodology: Mohammed El Amine Beyat And Mohammed El Aymene Beyat.

Investigation: Mohammed El Amine Beyat, Mohammed El Aymene Beyat, And Ahmed Korichi.

Discussion of results: Mohammed El Amine Beyat, Mohammed Kamel Benkaddour, Ahmed Korichi, Mohammed El Aymene Beyat And Abdessalam Moualdi.

Writing - Original Draft: Mohammed El Amine Beyat And Mohammed El Aymene Beyat.

Writing - Review and Editing: Mohammed El Amine Beyat, Mohammed Kamel Benkaddour, Ahmed Korichi, Mohammed El Aymene Beyat, And Abdessalam Moualdi.

Resources: Abdessalam Moualdi And Mohammed Kamel Benkaddour.

Supervision: Mohammed Kamel Benkaddour And Ahmed Korichi.

Approval of the final text: Mohammed El Amine Beyat, Mohammed Kamel Benkaddour, Ahmed Korichi, Mohammed El Aymene Beyat And Abdessalam Moualdi.

VIII. ACKNOWLEDGMENTS

The authors would like to express their sincere gratitude to the Laboratory of Artificial Intelligence and Information Technologies (LINATI) at the University of Kasdi Merbah (Ouargla, Algeria) for providing the specialized research environment and high-performance computational resources essential for this study. Furthermore, the authors acknowledge the support of the Laboratory of Electrical Engineering (LAGE) in facilitating the development and validation of the proposed framework.

IX. REFERENCES

- [1] B. Liu, N. Lv, Y. Guo, and Y. Li, "Recent advances on federated learning: A systematic survey", *Neurocomputing*, vol. 597, p. 128019, 2024. <https://doi.org/10.1016/j.neucom.2024.128019>
- [2] T. H. Rafi, F. A. Noor, T. Hussain, and D.-K. Chae, "Fairness and privacy preserving in federated learning: A survey", *Information Fusion*, vol. 105, p. 102198, 2024. <https://doi.org/10.1016/j.inffus.2023.102198>
- [3] S. Jenko, E. Papadopoulou, V. Kumar, S. S. Overman, K. Krepelkova, J. Wilson, E. L. Dunbar, C. Spice, and T. Exarchos, "Artificial intelligence in healthcare: How to develop and implement safe, ethical and trustworthy AI systems", *AI*, vol. 6, no. 6, p. 116, 2025. <https://doi.org/10.3390/ai6060116>
- [4] R. Marin Machado de Souza, A. Holm, M. Biczysk, and L. N. de Castro, "A systematic literature review on the use of federated learning and bioinspired computing", *Electronics*, vol. 13, no. 16, p. 3157, 2024. <https://doi.org/10.3390/electronics13163157>
- [5] K. Pan, Y.-S. Ong, M. Gong, H. Li, A. K. Qin, and Y. Gao, "Differential privacy in deep learning: A literature survey", *Neurocomputing*, vol. 589, p. 127663, 2024. <https://doi.org/10.1016/j.neucom.2024.127663>
- [6] I. Zhou, F. Tofigh, M. Piccardi, M. Abolhasan, D. Franklin, and J. Lipman, "Secure multi-party computation for machine learning: A survey", *IEEE Access*, vol. 12, pp. 53881–53899, 2024. <https://doi.org/10.1109/ACCESS.2024.3388992>
- [7] J. Yuan, W. Liu, J. Shi, and Q. Li, "Approximate homomorphic encryption based privacy-preserving machine learning: A survey", *Artificial Intelligence Review*, vol. 58, no. 3, p. 82, 2025. <https://doi.org/10.1007/s10462-024-11076-8>
- [8] J. Zhang, X. Cheng, L. Yang, J. Hu, X. Liu, and K. Chen, "SoK: Fully homomorphic encryption accelerators", *ACM Computing Surveys*, vol. 56, no. 12, pp. 1–32, 2024. <https://doi.org/10.1145/3676955>
- [9] V. Pais, S. Rao, B. Muniyal, and S. Yun, "Federated learning using Flower framework for enhanced medication safety in intensive care units", *International Journal of Data Science and Analytics*, vol. 20, no. 8, pp. 7039–7053, 2025. <https://doi.org/10.1007/s41060-025-00877-x>
- [10] J. Yang, R. Shi, D. Wei, Z. Liu, L. Zhao, B. Ke, H. Pfister, and B. Ni, "MedMNIST v2: A large-scale lightweight benchmark for 2D and 3D biomedical image classification", *Scientific Data*, vol. 10, no. 1, p. 41, 2023. <https://doi.org/10.1038/s41597-022-01721-8>
- [11] J. Xia, P. Li, Y. Mao, and M. Wu, "PT-ADP: A personalized privacy-preserving federated learning scheme based on transaction mechanism", *Information Sciences*, vol. 669, p. 120519, 2024. <https://doi.org/10.1016/j.ins.2024.120519>
- [12] R. Aziz, S. Banerjee, S. Bouzeffrane, and T. Le Vinh, "Exploring homomorphic encryption and differential privacy techniques towards secure federated learning paradigm", *Future Internet*, vol. 15, no. 9, p. 310, 2023. <https://doi.org/10.3390/fi15090310>

- [13] F. Liu, Z. Zheng, Y. Shi, Y. Tong, and Y. Zhang, “A survey on federated learning: A perspective from multi-party computation”, *Frontiers of Computer Science*, vol. 18, no. 1, p. 181336, 2024. <https://doi.org/10.1007/s11704-023-3282-7>
- [14] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, and Y. Liu, “BatchCrypt: Efficient homomorphic encryption for cross-silo federated learning”, in *Proc. USENIX Annual Technical Conference (USENIX ATC 20)*, 2020, pp. 493–506. <https://dl.acm.org/doi/10.5555/3489146.3489179>
- [15] H. Fang and Q. Qian, “Privacy preserving machine learning with homomorphic encryption and federated learning”, *Future Internet*, vol. 13, no. 4, p. 94, 2021. <https://doi.org/10.3390/fi13040094>
- [16] Y. Pan, Z. Chao, W. He, Y. Jing, H. Li, and L. Wang, “FedSHE: Privacy preserving and efficient federated learning with adaptive segmented CKKS homomorphic encryption”, *Cybersecurity*, vol. 7, no. 1, p. 40, 2024. <https://doi.org/10.1186/s42400-024-00232-w>
- [17] H. Kurniawan and M. Mambo, “Homomorphic encryption-based federated privacy preservation for deep active learning”, *Entropy*, vol. 24, no. 11, p. 1545, 2022. <https://doi.org/10.3390/e24111545>
- [18] O. Rainio, J. Teuho, and R. Klén, “Evaluation metrics and statistical tests for machine learning”, *Scientific Reports*, vol. 14, no. 1, p. 6086, 2024. <https://doi.org/10.1038/s41598-024-56706-x>
- [19] S. Sathyanarayanan and B. R. Tantri, “Confusion matrix-based performance evaluation metrics”, *African Journal of Biomedical Research*, vol. 27, no. 4S, pp. 4023–4031, 2024. <https://doi.org/10.53555/AJBR.v27i4S.4345>
- [20] C. Miller, T. Portlock, D. M. Nyaga, and J. M. O’Sullivan, “A review of model evaluation metrics for machine learning in genetics and genomics”, *Frontiers in Bioinformatics*, vol. 4, p. 1457619, 2024. <https://doi.org/10.3389/fbinf.2024.1457619>
- [21] D. Chicco and G. Jurman, “The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation”, *BMC Genomics*, vol. 21, no. 1, p. 6, 2020. <https://doi.org/10.1186/s12864-019-6413-7>
- [22] H. Sun, J. Cui, Y. Shao, J. Yang, L. Xing, Q. Zhao, and L. Zhang, “A gastrointestinal image classification method based on improved Adam algorithm”, *Mathematics*, vol. 12, no. 16, p. 2452, 2024. <https://doi.org/10.3390/math12162452>
- [23] S. Abd El-Ghany, M. A. Mahmood, and A. Abd El-Aziz, “Adaptive dynamic learning rate optimization technique for colorectal cancer diagnosis based on histopathological image using EfficientNet-B0 deep learning model”, *Electronics*, vol. 13, no. 16, p. 3126, 2024. <https://doi.org/10.3390/electronics13163126>
- [24] H. Arshad, T. Abdullah, M. Rehman, A. Hussain, F. Kanwal, and M. Parveen, “Optimizer-aware fine-tuning of Whisper Small with low-rank adaption: An empirical study of Adam and AdamW”, *Information*, vol. 16, no. 11, p. 928, 2025. <https://doi.org/10.3390/info16110928>
- [25] A. Johnny and K. Madhusoodanan, “Dynamic learning rate in deep CNN model for metastasis detection and classification of histopathology images”, *Computational and Mathematical Methods in Medicine*, vol. 2021, p. 5557168, 2021. <https://doi.org/10.1155/2021/5557168>
- [26] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, *et al.*, “Searching for MobileNetV3”, in *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 1314–1324. <https://doi.org/10.1109/ICCV.2019.00140>
- [27] M. Tan and Q. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks”, in *Proc. International Conference on Machine Learning (ICML)*, 2019, pp. 6105–6114. <https://doi.org/10.48550/arXiv.1905.11946>
- [28] Y. Li, G. Yuan, Y. Wen, J. Hu, G. Evangelidis, S. Tulyakov, Y. Wang, and J. Ren, “EfficientFormer: Vision transformers at MobileNet speed”, *Advances in Neural Information Processing Systems*, vol. 35, pp. 12934–12949, 2022. <https://doi.org/10.48550/arXiv.2206.01191>
- [29] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition”, in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. <https://doi.org/10.1109/CVPR.2016.90>
- [30] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, “A ConvNet for the 2020s”, in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 11976–11986. <https://doi.org/10.1109/CVPR52688.2022.01167>