Heterogeneity-Aware Private Personalized Federated Learning for Medical Imaging via Contrastive Distillation

Nazmus Shakib Shadin[†], Xinyue Zhang[‡], Jingyi Wang[§], and Miao Pan[⋆]
[†,‡]Department of Computer Science, Kennesaw State University, Marietta, GA, 30060 USA
[§]Department of Computer Science, San Francisco State University, San Francisco, CA, 94132 USA
[⋆]Department of Electrical and Computer Engineering, University of Houston, Houston, TX, 77204 USA
[†]nshadin@students.kennesaw.edu, [‡]xzhang48@kennesaw.edu, [§]jingyiwang@sfsu.edu, [⋆]mpan2@uh.edu

***Abstract*—Federated learning (FL) enables collaborative model training across decentralized clients without sharing raw data. However, its deployment in real-world settings faces severe challenges, including data heterogeneity, system heterogeneity with device capability variance, communication overhead, and privacy concerns. A common limitation of existing work is the lack of consideration for the joint impact of these issues, ultimately leading to degraded performance. These limitations are particularly evident in healthcare, where data distribution differs across hospitals and patient groups, which makes a single global model often inadequate. In this paper, we propose PFedCDP for medical imaging, a unified framework for privacy-preserving, heterogeneity-aware personalized federated learning (PFL), which addresses these issues. PFedCDP introduces a device capability-aware client clustering mechanism that assigns appropriately scaled neural architectures to devices based on computational capacity, memory, battery, and network metrics. Personalized training is then guided by a mechanism that preserves locally important parameters while still incorporating global knowledge, ensuring that clients benefit from both personalization and collaboration. To have a generalized server model, we add a post-hoc refinement stage that strengthens representation robustness and effectively transfers diverse cluster knowledge. Furthermore, the integration of differential privacy and model quantization ensures privacy and communication efficiency. Experimental evaluations on mammography datasets demonstrate that PFedCDP outperforms state-of-the-art baselines, achieving robust personalization and privacy preservation under heterogeneous and non-IID conditions in real-world healthcare scenarios. Our implementation repository is publicly available at https://github.com/shadhin39/PFedCDP.**

***Index Terms*—Federated Learning, Client Clustering, Knowledge Distillation, Fisher Information, Contrastive Learning.**

I. Introduction

Federated learning (FL) has emerged as a transformative distributed machine learning paradigm that enables collaborative model training across decentralized devices while preserving data locality and privacy [1]. Unlike traditional centralized approaches that require raw aggregation of data, FL allows multiple participants to contribute to a shared model without exposing their sensitive information [2]. This property is particularly crucial in today's era of big data, where massive volumes of information are generated across diverse sources, and it directly addresses fundamental privacy concerns in modern AI applications [3]. This revolutionary approach has gained substantial attention across a wide range of applications, including smart homes [4], autonomous vehicles [5], and critical domains like healthcare [6]. In healthcare, FL is particularly valuable, as strict privacy regulations restrict data sharing across hospitals. It has been applied to medical imaging and disease prediction, where multi-hospital collaboration can substantially improve model performance [6]. However, clinical data are often highly non-independent and identically distributed (non-IID). For example, in mammography, variations in imaging devices, acquisition protocols, and patient demographics across hospitals introduce significant domain shifts [7]. This heterogeneity can degrade the performance of a single global model and highlights the need for PFL approaches that adapt to local patient populations while still leveraging shared knowledge. Such data heterogeneity is only one part of the challenge. In practice, FL deployment must also contend with the heterogeneous computational capabilities of edge devices, which create bottlenecks as clients differ widely in processing power, memory, battery life, and connectivity [8]. These difficulties are further compounded by communication constraints and the need for robust privacy guarantees in real-world deployments [9].

These challenges highlight that beyond addressing heterogeneity, effective FL must also incorporate personalization to ensure reliable performance across diverse clients, especially in sensitive domains such as healthcare [6]. Unlike generic FL strategies that enforce a single global model, personalization allows local models to retain parameters that are most relevant to their unique data distributions [10]. This is critical in medical imaging tasks such as mammography, where data distributions vary widely between hospitals due to differences in imaging equipment, acquisition protocols, and patient demographics. For example, rural or resource-limited hospitals often face a scarcity of annotated mammograms and may encounter class imbalance problems, while larger urban medical centers may have more diverse and better-curated datasets [11]. Without personalization, a global model fails to perform adequately in low-resource settings, which could increase inequalities in healthcare quality. Moreover,

because patient data are highly sensitive and subject to strict privacy regulations, centralizing medical images for model training is infeasible. Under these circumstances, PFL offers a compelling solution, allowing hospitals to collaboratively learn robust global representations while ensuring that each institution's local model is adapted to its specific patient and resource environment. Recent work [12], [13] such as Fisher information-based parameter selection provides a principled way to achieve this balance by preserving parameters critical to local tasks while incorporating global knowledge.

Although personalization improves model effectiveness across diverse clinical settings, deployment also requires strong privacy protections to ensure that sensitive patient information remains more secure during training. Privacy-preserving techniques in FL have evolved significantly, with differential privacy (DP) emerging as the gold standard for formal privacy preserving technique [14]. Recent advances combine DP with communication efficiency techniques such as quantization to reduce overhead while maintaining model utility [15]. However, balancing privacy preservation with model performance remains a critical challenge, particularly in heterogeneous environments where device capabilities and data distributions vary significantly [16].

Despite these advances, existing solutions often address individual challenges, failing to provide a unified framework that simultaneously handles device heterogeneity, data distribution variations, and privacy requirements. While prior studies [9], [17] have focused on privacy preservation and Dinh et al. [10] have investigated performance costs, the combined treatment of these two aspects remains unexplored.

To address these limitations, we introduce PFedCDP, a novel privacy-preserving heterogeneity-aware PFL framework. Our approach employs capability-based clustering to manage device heterogeneity, while personalization on the client side is achieved by using Fisher information to preserve locally important parameters and we also apply knowledge distillation (KD), where the server's classification layer acts as a teacher to refine client models. On the server side, we adopt a two-stage strategy: (a) iterative aggregation of feature extractors during federated rounds, and (b) a final, post-hoc model refinement phase that synergistically combines multi-teacher Knowledge Distillation (MTKD) [18], informed by Shapley values [19], with Supervised Contrastive Learning (SCL) [20] on a public dataset. To ensure practical viability, the framework integrates DP [14] and quantization [21] for robust privacy protection and communication efficiency. Our framework makes several key contributions, which are listed below:

- A capability-based client clustering mechanism that efficiently manages device heterogeneity by stratifying clients into computational capability clusters.
- A Fisher Information-guided personalization strategy that selectively preserves important local parameters while incorporating global knowledge, as well as employing KD where a server model guides the training of the client's classification layer.
- A privacy-preserving communication protocol that combines DP with quantization for enhanced security and efficiency.
- A post-hoc refinement process that leverages Shapley value-weighted MTKD with SCL to improve final server model robustness.

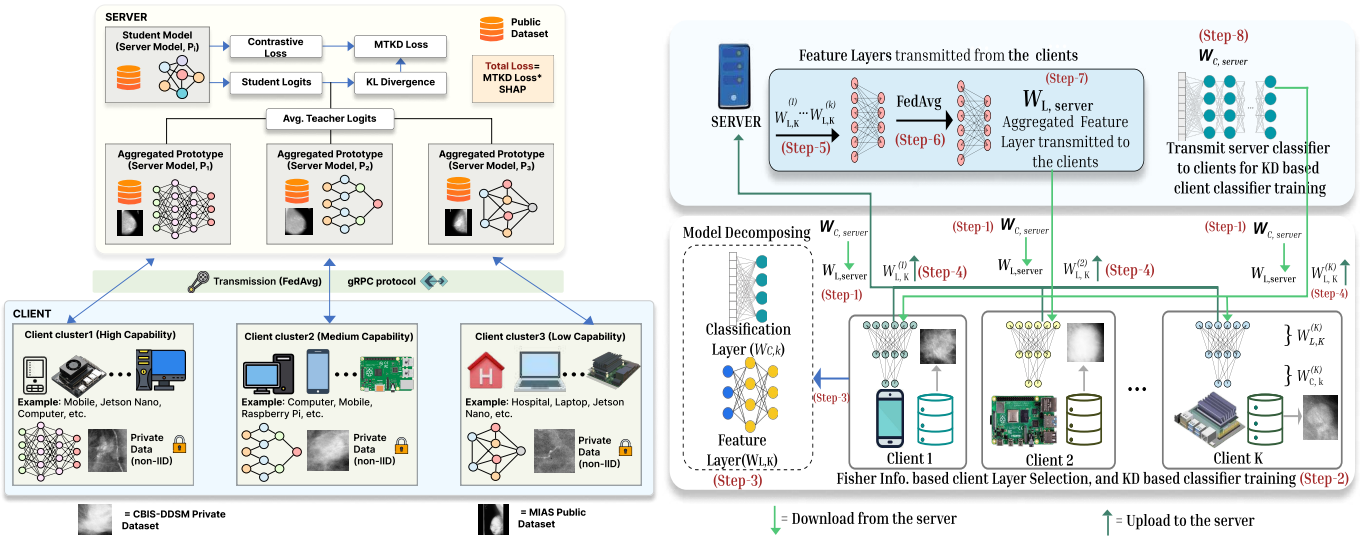
II. Literature Review

FL has emerged as a promising paradigm for decentralized machine learning, but its practical deployment remains constrained by non-IID data, heterogeneous devices, communication bottlenecks, and privacy risks [15]. Early solutions such as FedProx introduced proximal regularization to stabilize training under heterogeneity [16]. Another technique, pFedMe, applied a Moreau envelope framework to enable client-level adaptation with personalization [10]. Recent methods like PFedCS focus on personalization, which emphasizes model partitioning and similarity based collaboration, it also enhances knowledge sharing among clients with related classifiers [22]. For personalization, Fisher information-based methods have also been explored, as they preserve parameters that are most important to local tasks while still incorporating global knowledge [12]. These studies show that personalization is important for stable performance, especially in domains such as healthcare, since client distributions differ significantly there.

Another line of research focuses on KD as a mechanism for model heterogeneity and communication efficiency. FedDF pioneered ensemble distillation on the server to fuse various client models [8]. FedCKD extended this by combining multi-teacher guidance with personalized history to improve stability [23]. FedMD leveraged KD on shared public datasets to support heterogeneous client models, achieving an approximately 20% improvement in accuracy over standalone training [24]. KD has also been shown effective for handling device heterogeneity, where smaller models learn from larger or ensemble teachers using soft targets [25]. MTKD frameworks have been proposed to guide student models through multiple teachers, although they often remain sensitive to non-IID data distributions [18]. Multiple surveys underline KD's versatility in FL, spanning privacy preservation, generalization, and transfer across heterogeneous environments [25], [26].

Although KD methods primarily address model heterogeneity and communication efficiency, they often overlook fairness in contribution across clients, which Shapley value-based approaches tackle through systematic client contribution weighting. ShapleyFL applied Shapley values to client weighting for robust aggregation under adversarial or low-quality updates [4]. FedKDShap extended this principle to KD, emphasizing feature importance and mitigating non-IID degradation [19].

While Shapley value-based methods improve fairness and robustness in aggregation, other directions focus on strengthening feature representations to better handle non-IID data. FedRCL adapted SCL to prevent representation collapse,



(a) High-level overview showing PFedCDP architecture

(b) PFL training cycle within a client cluster and the server.

Fig. 1. The proposed PFedCDP framework architecture. (a) shows the high-level, end-to-end system, including heterogeneous client clusters on private dataset D_k and the server's post-hoc refinement stage on proxy dataset D_{pub}. (b) details the iterative training loop within a single client cluster and server, highlighting model decomposition for feature layer aggregation as well as Fisher information and KD based personalization.

strengthening global feature spaces while maintaining local adaptability [27].

In addition to representation-level improvements, the practical implementation of FL must also address privacy and communication constraints to ensure both security and efficiency. DP has become the widely accepted standard for formal guarantees [17], but its integration with PFL often degrades utility. DP-pFedDSU proposed dynamically sparsified client updates via reparameterization and adaptive norms, reducing the noise required under the same privacy budget and achieving stronger trade-offs between privacy and personalization [9].

III. Methodology

We propose PFedCDP, a novel PFL framework tailored to address real-world constraints in decentralized learning, including device heterogeneity, data heterogeneity, communication limitations, and privacy preservation. The overall framework is organized into two principal components: a client-side personalization phase and a server-side post-hoc refinement phase, both of them are described in detail in the subsequent subsections.

A. System Architecture

Figure 1 illustrates the proposed PFedCDP framework, which operates in a client–server environment with heterogeneous devices and a server-side refinement stage. The architecture is organized into two levels. The first one is the overview showing how heterogeneous client clusters connected to the central server using Figure 1a, and then the detailed training loop for a single-client cluster shown in Figure 1b.

On the client side, devices are grouped into three capability-based clusters (High, Medium, and Low) according to computational resources and network conditions. For example, high-capability devices, such as workstations or servers with GPUs, are assigned deeper neural networks; medium-capability devices such as personal laptops or desktops are given moderately complex models; and low-capability devices such as mobile phones or IoT nodes are assigned lightweight shallow models. This clustering ensures that weaker devices do not become bottlenecks while still allowing them to contribute effectively. For each client k, the client model W_k is decomposed into a feature extractor, and a classification layer ($W_{C,k}$). The feature extractor is responsible for learning general representations that can be shared and aggregated across clients, while the classification layer captures task-specific patterns specific to the client's local data. This framework keeps the classification layer on the local client side and only shares the feature extractor with the server, which ensures client specific personalization. It also allows each client to adapt their predictions to its own data distribution while still benefiting from global knowledge. On the server end, the feature extractors are aggregated using FedAvg to produce a prototype for each cluster. The server then refines these prototypes in a post hoc stage. In this stage, MTKD gathers knowledge from all cluster models. Then the Shapley value weights adjust the impact of each model. Finally, SCL strengthens the representation robustness using the public proxy dataset D_{pub}. This refinement enables the server to integrate diverse knowledge across clusters and improve generalization without requiring access to sensitive private data.

In healthcare applications such as mammography screening, this design is particularly beneficial because hospitals often vary in computational resources as well as in patient data distributions. PFedCDP ensures that hospitals with limited compute capacity or smaller datasets can still participate meaningfully without compromising privacy or accuracy.

Although client data are assumed to be non-IID in practice, in our experiments statistical heterogeneity is emulated using

a Dirichlet partitioning strategy [2].

B. Heterogeneity-Aware Client Clustering

To manage the device heterogeneity of client devices, we implement a capability-aware clustering mechanism prior to training. For each client k, four normalized capability parameters are computed to assess their suitability for FL. The *CPU capability* is calculated as $C_{CPU_k} = \frac{f_{current_k}}{f_{max_k}}$, which reflects how much of the device's maximum processing frequency is currently available. The *memory capability* is given by $C_{Mem_k} = \frac{M_{available_k}}{M_{total_k}}$, which indicates the proportion of usable memory relative to the total memory capacity, thereby capturing whether the client can accommodate training workloads. The *battery capability* is expressed as $C_{Battery_k} = \frac{\text{Battery Level}_k}{100}$, which directly translates the percentage of the battery into a normalized score that accounts for the reliability of the device under limited energy conditions. Finally, the *network capability* is defined as $C_{Net_k} = 1 - \frac{\text{current Latency}_k}{\text{maximum Acceptable Latency}}$, where higher latency values proportionally decrease the performance. This formulation make sure that clients with faster and more stable connections are assigned with higher capability, on the other hand those near the acceptable latency limit get lower scores because their communication is less efficient.

These four metrics provide a clear view of the computational, memory, energy, and communication resources of each client in a normalized form. Finally, all four matrices are combined into a one weighted overall score, C_{O_k}. The weights (w_1, w_2, w_3, w_4) can be tuned based on which resources matter most, and they can add up to 1 ($\sum w_i = 1$). For instance, if the battery level is considered less critical for a particular deployment, its weight could be reduced (e.g. $w_1 = 0.3, w_2 = 0.3, w_3 = 0.1, w_4 = 0.3$). If all resources matter the same, each one gets an equal weight of $1/4$. The score is calculated as follows:

$$C_{O_k} = w_1 C_{CPU_k} + w_2 C_{Mem_k} + w_3 C_{Battery_k} + w_4 C_{Net_k}. \tag{1}$$

Then clients are placed into one of the three clusters (High, Medium, Low) based on their overall score (C_{O_k}). This is relative to predefined thresholds (T_H, T_M).

$$\text{C}luster(C_{O_k}) = \begin{cases} \text{High}, & C_{O_k} \geq T_H \\ \text{Medium}, & T_M \leq C_{O_k} < T_H \\ \text{Low}, & C_{O_k} < T_M \end{cases} \tag{2}$$

We have organized our architecture into three distinct client clusters, where each cluster tailored to specific computational capabilities and corresponding neural network model complexities. Cluster 1 is designated as the high capability cluster. It utilizes Deep Neural Network models, and it is suitable for devices with substantial computational resources. The medium capability cluster (Cluster 2) will be assigned Medium Size Neural Network models, designed for devices with moderate processing power. Finally, the low capability cluster (Cluster 3) will employ very few layered neural networks, optimized for devices with limited computing capabilities.

C. Personalized Federated Training Loop

The core of our framework is an iterative training loop that combines client-side personalization with server-side aggregation of feature extractors. As described in Algorithm 1, each client first personalizes its feature extractor using Fisher information to preserve locally important parameters, while replacing the remaining parameters with the global knowledge received from the server. The client then refines its classification layer through KD guided by the server's model. After local updates are computed, clients apply differential privacy and quantization before transmitting their feature extractor updates to the server. Algorithm 2 then describes the server-side process, where client updates are dequantized, aggregated within each cluster, and subsequently refined through a post-hoc stage that integrates Shapley-weighted MTKD with SCL. Together, these two algorithms provide a step-by-step view of how PFedCDP balances personalization and generalization across heterogeneous clients.

1) Server-to-Client Model Distribution: Following the Figure 1b, the classification of clients into discrete clusters based on their computational capabilities, the iterative personalized training process is initiated. The first step in each communication round involves the distribution of the server-to-client model. At the start of each communication round t, the server sends a complete cluster-specific prototype model to each participating client k. This model consists of the aggregated feature extractor for that cluster from the previous round ($W^{t-1}_{L,global}$), and the server's own powerful classification layer for that cluster $W^{t-1}_{C,server}$.

2) Client-Side Personalization and Training: Upon receiving the server model ($W^{t-1}_{L,global}$ + $W^{t-1}_{C,server}$), each client k undertakes a multi-step local training process. The first step involves feature extractor personalization with fisher information. Fisher information, a classical statistical measure of parameter sensitivity, which captures how much each model parameter contributes to the likelihood of the observed data, making it a natural criterion for identifying which parameters are most important to preserve for local personalization [14]. In this stage, the client personalizes its local feature extractor $W_{L,k}$ by retaining parameters with high Fisher scores while allowing less critical parameters to be updated from the global model. For a parameter w_j, the fisher information is defined as

$$F(w_j) = \left(\frac{\partial \log L(w, D_k)}{\partial w_j} \right)^2, \tag{3}$$

where $L(w, D_k)$ denotes the likelihood function of the client's data. Based on a predefined threshold T_{Fisher}, two binary masks are created: a *Personal Mask* ($M_{personal}$) and a *Global Mask* (M_{global}). The two masks are used to separate parameters that should be preserved for local personalization from those that should be replaced with global updates. Parameters with high Fisher values, which contribute significantly to the client's local task, are preserved using the Personal Mask:

$$M_{personal}[j] = \begin{cases} 1, & \text{if } F(w_j) \geq T_{Fisher} \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

On the other hand, parameters with low Fisher values are deemed less critical for local specialization and are instead updated with the corresponding global parameters using the Global Mask:

$$M_{global}[j] = \begin{cases} 1, & F(w_j) \leq T_{Fisher} \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

The client's feature extractor for the current round, $W_{L,k}^t$, is then initialized by combining the preserved local parameters with the updated global parameters from the server's model $W_{L,global}^{t-1}$:

$$W_{L,k}^t = M_{personal} \odot W_{L,k}^{t-1} + M_{global} \odot W_{L,global}^{t-1}, \tag{6}$$

where $\odot$ denotes element-wise multiplication. This selective update strategy allows clients to maintain crucial local knowledge while benefiting from globally aggregated updates for less critical parameters. By following this strategy, the framework balances personalization and generalization.

After completion of the personalization step, the client trains its classification layer using KD. At this point, the personalized feature extractor layer $W_{L,k}^t$ remains frozen and the client trains and updates only its classification layer $W_{C,k}$. Here, the server model with its classification layer $((W_{L,global}^{t-1} + W_{C,server}^{t-1}))$ acts as the teacher and, on the other hand, the client model with its classification layer $(W_{C,k})$ serves as the student. Through this KD based teacher student setup, the server helps guide the client by aligning the client's predictions with the softened outputs of the global model, while the client also learns from its own local labels.

Formally, the student model θ_S generates logits $z_{student} = \theta_S(x)$ for local data x, while the teacher produces logits $z_{teacher}$ for the same inputs using $W_{L,global}^{t-1}$ and $W_{C,server}^{t-1}$. Both logits are converted into softened probability distributions using the softmax function σ with a distillation temperature T_{KD}:

$$\tilde{y}_{student} = \sigma(z_{student}/T_{KD}), \tag{7}$$

$$\tilde{y}_{teacher} = \sigma(z_{teacher}/T_{KD}). \tag{8}$$

The temperature T_{KD} smooths the distributions, enabling the student to capture relative class probabilities from the teacher rather than relying solely on hard labels.

The local training objective combines two components. First, the standard cross entropy loss checks how well the student model predicts the true labels y:

$$L_{CE} = -\frac{1}{|y_b|} \sum_{i \in b} \log P_{\theta_S}(y_i \mid x_i), \tag{9}$$

where y_b is the batch of true labels and $P_{\theta_S}(y_i \mid x_i)$ is the probability that the student predicts for each sample. Second, the Kullback-Leibler (KL) divergence measures how far the student's softened outputs are from the teacher's softened outputs:

$$L_{KD} = D_{KL}(\tilde{y}_{teacher} \,\|\, \tilde{y}_{student}). \tag{10}$$

The overall local training loss combines these two terms with weights that control their influence:

$$\mathcal{L}_{local} = (1 - \lambda) L_{CE} + \lambda T_{KD}^2 L_{KD}, \tag{11}$$

where λ controls how much weight the distillation term gets and T_{KD}^2 adjusts the KL part based on the temperature. By minimizing $\mathcal{L}_{local}$, the client updates its classifier to fit its own data while remaining consistent with the knowledge from the server's global model.

3) Client-to-Server Update Transmission: After completion of its local training, each client k prepares the updated model for transmission. The local model, $W_{local,k}^t$, is split into two distinct parts: first the updated feature extractor, $W_{L,k}^t$, and the other one is the updated classification layer, $W_{C,k}^t$. For aggregation on the server side, only the updated feature extractor, $W_{L,k}^t$, is going to be transmitted. This transmission strategy significantly reduces the communication cost, because the classifier stays on the local device, where it remains personalized to the client's own data. The feature extractor, conversely, represents more generalizable features learned across the client's dataset, which makes it suitable for global aggregation and integration into the server's prototype models. To protect user data and reduce network overhead, two techniques are applied concurrently after the client-side updates are computed and before transmission to the server: DP for privacy preservation and Quantization for communication efficiency.

i. Privacy Preservation with Differential Privacy: DP [15] is a widely used technique that protects sensitive data by adding calibrated noise to model updates. In our framework, this is achieved by privatizing the feature extractor update $\Delta W_{L,k}^t$ through L_2 norm clipping followed by the addition of Gaussian noise. C_2 is the clipping threshold that controls the maximum contribution of a client to the global update. Δ_{DP}^c is the local update after clipping, Δ_{DP}^t is the final local update after clipping and adding noise, $\mathcal{N}$ is the Gaussian noise added to ensure DP, and σ is the noise multiplier computed by the privacy accountant, and composition mechanism with respect to ϵ and δ [14].

$$\Delta_{DP}^c = \frac{\Delta W_{L,k}^t}{\max(1, \frac{||\Delta W_{L,k}^t||_2}{C_2})}, \tag{12}$$

$$\Delta_{DP}^t = \Delta_{DP}^c + \mathcal{N}(0, {C_2}^2 \sigma^2). \tag{13}$$

ii. Communication Efficiency via Quantization: To further enhance communication efficiency, the privatized update is quantized into a compact lower bit integer representation. Let ΔW_q^t denote the quantized update, and b the number of bits used for quantization. The transformation is expressed as follows:

$$\Delta W_q^t = \text{round}\left(\frac{\Delta_{DP}^t - \Delta w_{min}}{\Delta w_{max} - \Delta w_{min}} \times (2^b - 1) \right), \tag{14}$$

where Δw_{min} and Δw_{max} represent the minimum and maximum values of the update, respectively. This quantization step reduces the precision of transmitted updates, thereby lowering the bandwidth requirements, while maintaining sufficient fidelity for effective aggregation at the server.

4) Server-Side Aggregation: To construct a stronger global representation from distributed training, the server aggregates feature extractor updates collected from clients in each cluster. Since the transmitted updates are quantized for efficiency, the server first dequantizes them and then aggregates the results.

i. Dequantization: Since client updates are transmitted in compressed form to save bandwidth, the server reconstructs an approximate full precision update from each client k by dequantization (ΔW_{dq}^k) [21]:

$$\Delta W_{dq}^k = \Delta w_{min} + \frac{\Delta W_q^k}{2^b - 1} \times (\Delta w_{max} - \Delta w_{min}) \quad (15)$$

ii. Aggregation: The server aggregates the dequantized feature extractor updates using Federated Averaging (FedAvg) to produce the new global feature extractor for the next round, $W_{L,global}^t$ [3].

D. Post-Hoc Server-Side Refinement with SCL and Shapley Values informed MTKD

Following completion of federated training rounds, the server initiates a crucial post-hoc refinement phase for aggregated prototype models ($P_1, P_2, \ldots, P_n$) within the PFedCDP architecture. This phase leverages a public dataset and integrates both Ensemble MTKD [18] with Shapley values-based weighting and SCL [20]. Our motivation for integrating Shapley values-based weighting [19] and SCL stems from their efficacy in leveraging a local public or proxy dataset. This public dataset is separated from anything learned in the private dataset of all the clients. It lets the server refine the model to improve robustness and discrimination while without compromising the private data.

The server starts its refinement step by treating the cluster prototypes as a group of teacher models. Each prototype P_k effectively gathers the knowledge from its respective client cluster after the completion of all federated rounds. Then each prototype serves as an individual teacher model $\theta_T^{(k)}$. The server then initializes a new student model, θ_S, which will become the refined global model. A labeled proxy dataset $(x_{\text{train}}, y_{\text{train}})$ is used to combine the knowledge of all teachers.

Prior to training of θ_S, PFedCDP computes Shapley value-based feature importance for each teacher model $\theta_T^{(k)}$ using the proxy data. A feature importance vector $\psi^{(k)}$ is derived using shapley values, which quantifies each input feature's contribution to the teacher's predictions on a representative subset of class-balanced test samples. These individual $\psi^{(k)}$ vectors are then aggregated (e.g., averaged) to form a combined importance weight ψ, capturing the collective feature importance across the teacher ensemble. This ψ will subsequently guide the distillation process by weighting the loss function [19],

$$\psi = \frac{1}{K} \sum_{k=1}^{K} \psi^{(k)}. \quad (16)$$

The training of the student model θ_S proceeds using ensemble MTKD over multiple epochs (E) and mini-batches (x_b, y_b) from the public training data. For each batch, we have followed the two steps which are: i) Each teacher $\theta_T^{(k)}$ generates logits $z_T^{(k)} = \theta_T^{(k)}(x_b)$. These are combined to form an ensemble prediction of the teacher $\bar{z}_T = \frac{1}{K} \sum_{k=1}^{K} z_T^{(k)}$. ii) Softened probability distributions are derived from teacher and student model logits using a distillation temperature using equation 7, and 8. The KD loss (L_{KD}) is then computed as the KL divergence between these softened distributions using equation 10. Crucially, instead of traditional cross-entropy loss, our framework integrates SCL as the primary classification objective. SCL enhances representation learning by promoting intraclass compactness (pulling samples from the same class together) and interclass separability (pushing samples from different classes apart) [20]. This is achieved by generalizing the standard contrastive loss to handle an arbitrary number of positive examples for a given anchor within a mini-batch. This method encourages the encoder to generate closely aligned representations for all instances of the same class, leading to a more robust clustering of the representation space. The supervised contrastive loss for a single anchor sample i is

Algorithm 1 PFedCDP: Client-Side Local Training

Input: Cluster index c, previous global feature extractor $W_{L,global}^{t-1}(c)$, server's classifier $W_{C,server}^{t-1}(c)$, local data D_k, Number Local Epochs E.
Output: Quantized update ΔW_q^t for feature extractor
Download from the server:
    Global feature extractor $W_{L,global}^{t-1}(c)$.
    Server classifier $W_{C,server}^{t-1}(c)$.
Compute Fisher information $F(w_j)$ for each parameter w_j using eq. 3.
Build masks $M_{personal}$ using eq. 4 and $M_{global}[j]$ using eq. 5.
Combine parameters: $W_{L,k}^t \leftarrow M_{personal} \odot W_{L,k}^{t-1} + M_{global} \odot W_{L,global}^{t-1}$ using eq. 6.
Freeze the updated feature extractor $W_{L,k}^t$.
Initialize student model: feature extractor $W_{L,k}^t$, classifier $W_{C,k}$. Teacher model: $W_{L,global}^{t-1} + W_{C,server}^{t-1}$.
for each epoch $e = 1$ to E **do**
    Compute student logits $z_{student}$ and teacher logits $z_{teacher}$ for batch $(x, y) \subset D_k$.
    $\tilde{y}_{student} \leftarrow \sigma(z_{student}/T_{KD})$, using eq. 7.
    $\tilde{y}_{teacher} \leftarrow \sigma(z_{teacher}/T_{KD})$ using eq. 8.
    Compute $L_{CE}, L_{KD}, \mathcal{L}_{local}$ using eq. 9, 10, 11.
    Update local classifier $W_{C,k}$ by minimizing $\mathcal{L}_{local}$
end for
Compute raw update: $\Delta W_{L,k}^t \leftarrow W_{L,k}^t - W_{L,global}^{t-1}$.
Clip and add noise (Differential Privacy):
ΔW_{DP}^c; ΔW_{DP}^t. using eq. 12 and eq. 13 respectively.
Quantize update: ΔW_q^t using eq. 14.
Return: Transmit ΔW_q^t to the server.

given by:

$$\mathcal{L}_i^{SCL} = \frac{-1}{2N_{\tilde{y}_i} - 1} \sum_{j=1, i \neq j, \tilde{y}_i = \tilde{y}_j}^{2N} \log \frac{\exp(z_i \cdot z_j / \tau)}{\sum_{k=1, i \neq k}^{2N} \exp(z_i \cdot z_k / \tau)}. \tag{17}$$

Here, z_i, z_j, and z_k are the representation embeddings of the samples, the sum in the numerator is over all other "positive" samples j in the mini-batch that share the same label as the anchor i, $N_{\tilde{y}_i}$ is the total number of images in the mini-batch that have the same label, $\tilde{y}_i$, as the anchor, and τ is a temperature hyperparameter. This $\mathcal{L}^{SCL}$ term, summed over all samples in the batch, directly contributes to the overall loss.

The total loss $\mathcal{L}_{SupMTKD}$ for the student model combines the SCL loss and the KD loss:

$$\mathcal{L}_{SupMTKD} = (1 - \alpha)\mathcal{L}^{SCL} + \alpha T^2 L_{\text{KD}}. \tag{18}$$

Here, α is a balancing hyperparameter, and T^2 scales the KL term. Finally, $\mathcal{L}_{SupMTKD}$ is weighted by the Shapley-based importance ψ_i to yield the feature-weighted distillation loss $\mathcal{L}_{SupMTKDShap}$:

$$\mathcal{L}_{SupMTKDShap} = \frac{1}{|F|} \sum_{i=1}^{|F|} \psi_i \mathcal{L}_{SupMTKD}. \tag{19}$$

The overall server-side refinement process, which combines MTKD, Shapley weighting, and SCL, is shown in Algorithm 2.

IV. Performance Evaluation

This section discusses the experimental evaluation of our proposed PFedCDP framework. We assess its performance in handling data and system heterogeneity, preserving privacy, and achieving high model accuracy, precision, recall, f1-score, specificity, and AUC compared to baseline methods [28].

A. Experimental Setup

1) Datasets and Distribution: To simulate a realistic medical imaging scenario, we utilize two public mammography datasets: the Curated Breast Imaging Subset of DDSM (CBIS-DDSM) [29] and the Mammographic Image Analysis Society (MIAS) dataset [30].

The private data held by clients is represented by the CBIS-DDSM dataset. This is a curated and standardized version of the Digital Database for Screening Mammography (DDSM), which consists of 2,620 scanned film mammography studies. It is a comprehensive collection containing normal, benign, and malignant cases with verified pathology information. We use 2,326 training images and 772 testing images for our experiments. To simulate statistical heterogeneity, we split the training data among all clients using a dirichlet distribution with ($\alpha = 0.5$). This produces non IID data at each client side. This setting reflects real world scenarios where hospitals hold patient data with different biases.

The MIAS dataset serves as the public data on the server. It contains mammography images with labels that mark each case as Benign (B) or Malignant (M). We used 280 training samples and 50 testing samples for the refinement stage. During this step, SCL and Shapley weighted MTKD help to strengthen the generalization of the global model without requiring any private client data.

2) Simulation Environment and Models: Our simulation uses the Flower framework [2] to handle communication between the central server and the heterogeneous client environment. We simulate the three client clusters defined in our methodology (High, Medium, and Low capability). Each cluster is assigned a different model architecture that matches its computational capability. We utilized the Flower framework's gRPC-based architecture to reliably simulate client-server interactions, with each cluster communicating on a dedicated port (e.g., 8080, 8081, 8082). The framework employs three CNN models on the client side that are tailored to the device capability. Model A is for high-capability devices that uses two convolutional blocks with 32 and 64 filters, followed by a dense layer of 128 units prior to the classification layer. Then, Model B is designed for medium-capability devices, which have lighter convolutional layers with 16 and 32 filters,

Algorithm 2 PFedCDP: Server-Side Training and Post-Hoc Refinement

Input: Client Clusters C, thresholds (T_H, T_M), rounds T, Training Epoch E, proxy dataset $(x_{\text{train}}, y_{\text{train}})$
Output: Refined global student model θ_S
for each round $t = 1$ to T **do**
  Send $(W_{L,global}^{t-1}(c), W_{C,server}^{t-1}(c))$ to clients in all c
  Receive client's local updates quantized ΔW_q^k
  for each client k **do**
    Dequantize: $\Delta W_{dq}^k \leftarrow \text{DeQuant}(\Delta W_q^k)$ using eq. 15.
  end for
  for each cluster c **do**
    $\Delta W_{agg}^c \leftarrow \text{Avg}_{k \in C(c)}(\Delta W_{dq}^k)$ $(FedAvg)$
    $W_{L,global}^t(c) \leftarrow W_{L,global}^{t-1}(c) + \Delta W_{agg}^c$
  end for
end for
Broadcast ΔW_{agg}^c, $W_{L,global}^t(c)$ to the client clusters.
Collect final cluster models $\{P_1, \ldots, P_C\}$
Compute Shapley feature importance vectors $\{\psi^{(1)}, \ldots, \psi^{(C)}\}$ and average to get ψ using eq. 16.
Initialize student model θ_S
for each training epoch 1 to E **do**
  for each batch (x_b, y_b) in proxy dataset $(x_{\text{train}}, y_{\text{train}})$ **do**
    Compute teacher logits $\{z_T^{(k)} = P_k(x_b)\}_{k=1}^C$ and average to get $\bar{z}_T$
    Compute student logits $z_S \leftarrow \theta_S(x_b)$
    Compute distillation loss L_{KD} using eq. 10 and $\mathcal{L}_{SCL}$ using embeddings of θ_S with eq. 17.
    Calculate $\mathcal{L}_{SupMTKD}$ using eq. 18.
    Apply Shapley weighting: $\mathcal{L}_{SupMTKDShap} \leftarrow \psi \cdot \mathcal{L}_{SupMTKD}$ using eq. 19.
    Update θ_S with $gradientdescent$
  end for
end for
Return: Refined model θ_S

a 64-unit dense layer, and includes dropout regularization. Finally, Model C is built for low-capability devices, which is an ultralight version with only 8 and 16 filters, and a 32-unit dense layer. For post-hoc refinement on the server, a student model is trained with 16 and 32 filters, a 64-unit dense layer, and a dropout of 0.5, ending with a softmax classification layer. The simulation runs for 20 federated rounds, with each client performing local training for 20 epochs per round, tailored to its capability. For privacy and efficiency, we apply Differential Privacy ($\epsilon = 5.0$, $C_2 = 2.5$) and Quantization (8 bits) to the model updates before transmission to the server.

B. Evaluation

We have evaluated PFedCDP against two baselines: a centralized model trained on all private data (serving as an upper bound) and a standard FedAvg implementation. We use a wide set of metrics to evaluate the performance of the dataset given the medical nature of the datasets. We report Accuracy, Precision, Recall (Sensitivity), Specificity, F1 Score, and AUC (Area Under the Curve), as the clinical impact of false negatives and false positives is really important in these scenarios.

1) Quantitative Analysis: Tables I and II show the final test results. PFedCDP clearly improves over FedAvg on every metric for both private (CBIS-DDSM) and public (MIAS) datasets. In the more challenging CBIS-DDSM dataset, PFedCDP reaches an accuracy of 94.1% and a high sensitivity of 93.5%. These results get close to the centralized upper bound and show strong performance on non IID data. On MIAS, the model reaches near perfect numbers, including 99.5% accuracy and a 99.9% AUC. This highlights how effective the post hoc refinement stage is at producing a well generalized global model.

2) Heterogeneity and Personalization Analysis: Figure 2 compares the performance of heterogeneous client clusters on the CBIS-DDSM dataset when training with 10, 50 and 100 clients. Cluster A (high capability) consistently achieves the highest accuracy across all settings. In contrast, Clusters B and C also show steady improvement. It demonstrates that, our framework enables clients of varying capacities to benefit from federated training. In particular, with the fewer clients (e.g., 10 clients) the framework achieves higher final accuracy and faster convergence than 50 or 100 clients. This is because each client holds more samples, which produces less noise to local gradients and enables more effective personalization. As the number of clients increases, data fragmentation and client drift become more pronounced. This leads to slower convergence and slightly lower peak accuracy. However, in all settings, Fisher information based personalization allows clients to retain critical local knowledge while still integrating global updates. Thus, the architecture mitigates the adverse effects of non-IID data and heterogeneous device capacities.

3) Post-Hoc Refinement Evaluation: Figure 3 shows the effectiveness of the server side post-hoc refinement stage. It compares the final PFedCDP student model with FedAvg and a centralized model across six performance metrics on the MIAS test dataset. In every plot in the corresponding subfigures, the PFedCDP model, which is refined with Shapley weighted MTKD and SCL, converges faster and reaches higher final scores than FedAvg. This ensures that the refinement stage effectively gathers knowledge from the different teacher models and builds a stronger and more discriminative global model. As a result, it generalizes better on the public/proxy dataset.

4) Comparison with State-of-the-Art (SOTA): To better understand its performance, we also compare PFedCDP with several recent state of the art (SOTA) approaches. Tables III and IV show that PFedCDP demonstrates highly competitive, and superior performance in both public and private datasets, similar to our approach.

In summary, PFedCDP effectively handles heterogeneity and privacy in FL through capability-aware clustering, Fisher-guided personalization, and advanced post-hoc refinement which delivers a robust, scalable, and high-performing PFL framework.

V. Conclusion

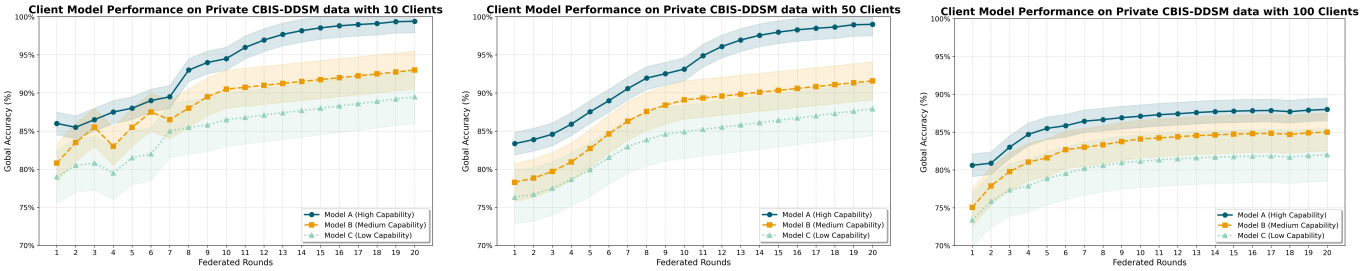
This paper introduces PFedCDP, a novel framework that addresses the key challenges of FL data heterogeneity, system heterogeneity, privacy, and communication overhead. Our approach provides a practical solution to train personalized models on diverse devices with limited resources. PFedCDP uniquely combines capability-aware client clustering, Fisher information-guided personalization, and a server-side refinement stage using Shapley-weighted distillation and contrastive learning. For security and efficiency, the framework also integrates differential privacy and quantization. Our comprehensive experimental evaluations on public mammography datasets show that PFedCDP outperforms FedAvg, achieving 94.1% accuracy on the private CBIS-DDSM data and 99.5% on the public MIAS data. These results validate that our integrated approach effectively mitigates the challenges of non-IID data and system variance. For future work, we plan to explore more dynamic client clustering algorithms and investigate the framework's applicability to other data modalities, such as text and time series data. In general, PFedCDP marks a step toward practical, and efficient PFL systems, especially in healthcare.

TABLE I
PERFORMANCE COMPARISON ON THE CBIS-DDSM (PRIVATE) DATASET.

Method	Accuracy(%)	Precision(%)	Recall/Sensitivity(%)	F1-Score(%)	Specificity(%)	AUC(%)
Centralized	96.2	96.6	95.8	96.2	96.5	98.8
Standard FedAvg	85.7	86.5	84.0	85.2	87.0	91.0
PFedCDP (Ours)	**94.1**	**94.8**	**93.5**	**94.1**	**94.5**	**97.5**

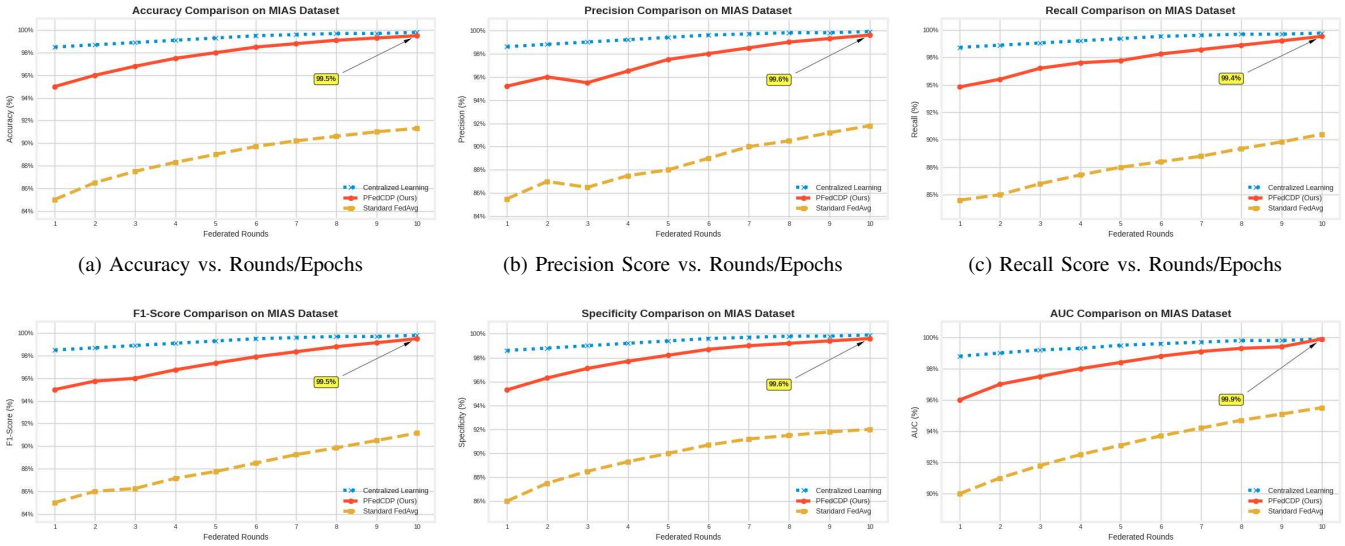
TABLE II
PERFORMANCE COMPARISON ON THE MIAS (PUBLIC/PROXY) DATASET.

Method	Accuracy(%)	Precision(%)	Recall/Sensitivity(%)	F1-Score(%)	Specificity(%)	AUC(%)
Centralized	99.8	99.9	99.7	99.8	99.9	99.9
Standard FedAvg	91.3	91.8	90.5	91.1	92.0	95.5
PFedCDP (Ours)	**99.5**	**99.6**	**99.4**	**99.5**	**99.6**	**99.9**



(a) No. of Clients - 10 (accuracy) (b) No. of Clients - 50 (accuracy) (c) No. of Clients - 100 (accuracy)

Fig. 2. Global accuracy of heterogeneous client clusters (A: High capability, B: Medium capability, C: Low capability) on non-IID partitions of the CBIS-DDSM dataset across 20 federated rounds. With fewer clients (10 clients), larger local datasets yield faster convergence and higher accuracy, while 50 and 100 clients introduce more fragmentation and drift, reducing peak accuracy. Nevertheless, all clusters show steady improvement, demonstrating that PFedCDP allows clients of varying capabilities to benefit from both global aggregation and Fisher Information guided personalization.



(a) Accuracy vs. Rounds/Epochs (b) Precision Score vs. Rounds/Epochs (c) Recall Score vs. Rounds/Epochs

(d) F1-Score vs. Rounds/Epochs (e) Specificity Score vs. Rounds/Epochs (f) AUC Score vs. Rounds/Epochs

Fig. 3. performance comparison on the public MIAS dataset. The subfigures illustrate the better performance of the final student model (enhanced via post-hoc refinement) over the standard FedAvg baseline across six key metrics: (a) Accuracy, (b) Precision, (c) Recall, (d) F1-Score, (e) Specificity, and (f) AUC.

ACKNOWLEDGMENTS

This work was supported in part by the U.S. National Science Foundation under Grants CNS-2107057, CNS-2318664, CSR-2403249, CNS-2431596, CNS-2431594, NSF-2348417 and NSF-2431597.

REFERENCES

[1] P. D. Lam, V. P. Tinh, D.-D. Le, N. H. Nam, T. A. Khoa *et al.*, "Joint federated learning using deep segmentation and the gaussian mixture model for breast cancer tumors," *IEEE Access*, vol. 12, pp. 94 231–94 249, 2024.

[2] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, J. Fernandez-Marques, Y. Gao, L. Sani, K. H. Li, T. Parcollet, P. P. B. de Gusmão *et al.*, "Flower: A friendly federated learning research framework," *arXiv preprint arXiv:2007.14390*, 2020.

[3] X. Zhang, R. Chen, J. Wang, H. Zhang, and M. Pan, "Energy efficient federated learning over cooperative relay-assisted wireless networks," in *GLOBECOM 2022-2022 IEEE Global Communications Conference*. IEEE, 2022, pp. 179–184.

[4] Q. Sun, X. Li, J. Zhang, L. Xiong, W. Liu, J. Liu, Z. Qin, and K. Ren, "Shapleyfl: Robust federated learning based on shapley value," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 2096–2108.

[5] S. R. Pokhrel and J. Choi, "Federated learning with blockchain for autonomous vehicles: Analysis and design challenges," *IEEE Transactions on Communications*, vol. 68, no. 8, pp. 4734–4746, 2020.

TABLE III
PERFORMANCE COMPARISON ON THE CBIS-DDSM DATASET

Reference and Methodology	Acc. (%)
PFedCDP (Ours) *PFL with Post-Hoc Refinement*	**94.1**
Joint FL [1] *Joint FL with Gaussian Mixture Model*	93.1
Optimized FL [11] *FL with Marine Predators Algorithm*	92.32
Transfer Learning FL [31] *FedAvg with CNN*	89.33
FL for Enhanced Deep Learning [7] *Deep Learning in FL*	75.0
Data Management App for FL [32] *Basic FL Application*	71.0

TABLE IV
PERFORMANCE COMPARISON ON THE MIAS DATASET

Reference and Methodology	Acc. (%)
PFedCDP (Ours) *PFL with Post-Hoc Refinement*	**99.5**
Screening CNN [33] *CNN on Screening Mammography*	99.14
Ensemble Classifier [34] *Optimized Ensemble Method*	97.76
Joint FL [1] *Joint FL with Gaussian Mixture Model*	96.7
Image Segmentation [35] *Pre-processing and Segmentation*	96.22
Comparative Analysis [36] *CNN*	94.23

[6] D. C. Nguyen, Q.-V. Pham, P. N. Pathirana, M. Ding, A. Seneviratne, Z. Lin, O. Dobre, and W.-J. Hwang, "Federated learning for smart healthcare: A survey," *ACM Computing Surveys (Csur)*, vol. 55, no. 3, pp. 1–37, 2022.

[7] D. G. Patadia, "Federated learning for enhanced deep learning integration," Ph.D. dissertation, Institute of Technology, 2024.

[8] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," *Advances in neural information processing systems*, vol. 33, pp. 2351–2363, 2020.

[9] C. Wang, Y. Zhang, N. Gao, and Q. Luo, "Differential privacy personalized federated learning based on dynamically sparsified client updates," *arXiv preprint arXiv:2503.09192*, 2025.

[10] C. T Dinh, N. Tran, and J. Nguyen, "Personalized federated learning with moreau envelopes," *Advances in neural information processing systems*, vol. 33, pp. 21 394–21 405, 2020.

[11] J. K. Mishra and A. A. Sharma, "Optimized federated learning algorithm for breast cancer detection using the marine predators algorithm," *Jes*, vol. 20, pp. 911–20, 2024.

[12] J. Liu, J. Ren, R. Jin, Z. Zhang, Y. Zhou, P. Valduriez, and D. Dou, "Fisher information-based efficient curriculum federated learning with large language models," *arXiv preprint arXiv:2410.00131*, 2024.

[13] D. Jhunjhunwala, S. Wang, and G. Joshi, "Fedfisher: Leveraging fisher information for one-shot federated learning," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2024, pp. 1612–1620.

[14] X. Yang, W. Huang, and M. Ye, "Dynamic personalized federated learning with adaptive differential privacy," *Advances in Neural Information Processing Systems*, vol. 36, pp. 72 181–72 192, 2023.

[15] X. Zhang, J. Ding, M. Wu, S. T. Wong, H. Van Nguyen, and M. Pan, "Adaptive privacy preserving deep learning algorithms for medical data," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2021, pp. 1169–1178.

[16] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[17] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318.

[18] H. Zhang, D. Chen, and C. Wang, "Confidence-aware multi-teacher knowledge distillation," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 4498–4502.

[19] N. S. Shadin and X. Zhang, "Fedkdshap: Enhancing federated learning via shapley values driven knowledge distillation on non-iid data," in *Companion Proceedings of the ACM on Web Conference 2025*, 2025, pp. 1744–1751.

[20] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," *Advances in neural information processing systems*, vol. 33, pp. 18 661–18 673, 2020.

[21] N. Shlezinger, M. Chen, Y. C. Eldar, H. V. Poor, and S. Cui, "Federated learning with quantization constraints," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 8851–8855.

[22] S. Wu, Y. Jia, B. Liu, H. Xiang, X. Xu, and W. Dou, "Pfedcs: A personalized federated learning method for enhancing collaboration among similar classifiers," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 20, 2025, pp. 21 572–21 580.

[23] P. Wang, B. Liu, W. Guo, Y. Li, and S. Ge, "Towards personalized federated learning via comprehensive knowledge distillation," in *2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2024, pp. 3807–3812.

[24] D. Li and J. Wang, "Fedmd: Heterogenous federated learning via model distillation," *arXiv preprint arXiv:1910.03581*, 2019.

[25] L. Qin, T. Zhu, W. Zhou, and P. S. Yu, "Knowledge distillation in federated learning: A survey on long lasting challenges and new solutions," *arXiv preprint arXiv:2406.10861*, 2024.

[26] A. Mora, I. Tenison, P. Bellavista, and I. Rish, "Knowledge distillation for federated learning: a practical guide," *arXiv preprint arXiv:2211.04742*, 2022.

[27] S. Seo, J. Kim, G. Kim, and B. Han, "Relaxed contrastive learning for federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 12 279–12 288.

[28] N. S. Shadin, S. Sanjana, S. Chakraborty, and N. Sharmin, "Performance analysis of glaucoma detection using deep learning models," in *2022 International conference on innovations in science, engineering and technology (ICISET)*. IEEE, 2022, pp. 190–195.

[29] R. S. Lee, F. Gimenez, A. Hoogi, K. K. Miyake, M. Gorovoy, and D. L. Rubin, "A curated mammography data set for use in computer-aided detection and diagnosis research," *Scientific data*, vol. 4, no. 1, pp. 1–9, 2017.

[30] J. Suckling, "The mammographic images analysis society digital mammogram database," in *Exerpta Medica. International Congress Series, 1994*, vol. 1069, 1994, pp. 375–378.

[31] Y. N. Tan, V. P. Tinh, P. D. Lam, N. H. Nam, and T. A. Khoa, "A transfer learning approach to breast cancer classification in a federated learning framework," *IEEE Access*, vol. 11, pp. 27 462–27 476, 2023.

[32] D. Tkachenko and M. Mazur-Milecka, "A mammography data management application for federated learning," in *2024 16th International Conference on Human System Interaction (HSI)*. IEEE, 2024, pp. 1–6.

[33] S. Acosta-Jiménez, S. A. Gonzalez-Chavez, J. Camarillo-Cisneros, C. Pacheco-Tena, M. Barcenas-Lopez, L. E. Gonzalez-Lozada, C. Hernandez-Orozco, J. H. Burboa-Delgado, and R. E. Ochoa-Albiztegui, "Preliminary results: Comparison of convolutional neural network architectures as an auxiliary clinical tool applied to screening mammography in mexican women," *Journal of medical and biological engineering*, vol. 44, no. 3, pp. 390–400, 2024.

[34] R. Laishram and R. Rabidas, "An optimized ensemble classifier for mammographic mass classification," *Computers and Electrical Engineering*, vol. 119, p. 109488, 2024.

[35] T. Soomro, K. M. Mehdar, A. Ali, F. B. Ubaid, M. Irfan, S. E. M. Elshafie, A. M. Mashraqi, A. A. Asiri, N. H. M. Khalid, and H. T. Halawani, "A computational model for enhanced mammographic image pre-processing and segmentation," *CMES Computer Modeling in Engineering and Sciences*, pp. 1–42, 2025.

[36] F. H. Alhsnony and L. Sellami, "Advancing breast cancer detection with convolutional neural networks: a comparative analysis of mias and ddsm datasets," in *2024 IEEE 7th international conference on advanced technologies, signal and image processing (ATSIP)*, vol. 1. IEEE, 2024, pp. 194–199.