Cluster Domain-Aware Client Selection for Federated Learning in The Healthcare Field (CDCSF)

Sanaa Lakrouni, Marouane Sebgui, Slimane Bah
Smart Communication Research Team, Mohamedia School of Engineers,
University Mohammed V, Rabat, Morocco

***Abstract*—Client selection remains a critical challenge in Federated Learning (FL). Resource-aware strategies aim to reduce training delays and mitigate stragglers by selecting an appropriate subset of clients in each round. However, these methods prioritize computationally strong clients and exclude resource-constrained clients. In healthcare settings, this approach is impractical because it removes entire domains from training, which harms generalisation. To address these challenges, we propose CDCSF, a domain-aware client selection framework that re-partitions clients into domain-homogeneous groups in each iteration. CDCSF is a dynamic clustering framework based on the (EM) algorithm that clusters clients based on local feature prototypes to enhance domain diversity. The framework incorporates a reliability score derived from an exponential moving average of training time to favor efficient clients. Simultaneously, a fairness score is introduced to ensure that underrepresented clients can still contribute to the training. This approach preserves sufficient representation across all domains to improve model generalization and accelerate convergence. We conduct extensive experiments on a healthcare benchmark dataset to validate the effectiveness of CDCSF. The proposed method improves accuracy by 2% over FedAvg under domain shift and outperforms PoC by 8%. With the proposed adaptive client selection strategy, we further demonstrate that CDCSF converges significantly faster than baseline methods under heterogeneous resource and data conditions.**

***Keywords*—Federated learning; healthcare; distributed learning; data heterogeneity**

I. Introduction

A variety of state-of-the-art deep learning models have demonstrated significant performance in the medical field. These traditional machine learning (ML) approaches rely on centralizing data at a central server for training. However, researchers have started to face challenges related to data privacy and insufficient data availability, which restrict the development of high-performing models in the healthcare field. Medical data are highly sensitive and, due to strict privacy regulations, this data cannot be shared with external entities. As a result, data from different institutions is stored on isolated local clients. This decentralization creates a major difficulty for the healthcare field. Recently, federated learning (FL) has been proposed to address privacy and efficiently leverage data from diverse and distributed sources [1]. FL trains a global model collaboratively, thereby keeping data private. Each client trains a local model and communicates only their local parameters to the server for global aggregation. Existing FL works have been successfully applied to brain tumor diagnosis [2], COVID-19 detection [3], and breast cancer classification [4] and achieved competitive performance to state-of-the-art centralized learning [5]. However, this decentralized paradigm suffers from performance degradation due to both resource constraints [6] and statistical heterogeneity. In conventional FL methods, most works rely on synchronous aggregation. In synchronous setups, the server cannot proceed until it collects all updates from every participant. This makes FL performance dependent on the client-side delays caused by slow local training or communication latency. To overcome these challenges, traditional client selections (CS) have been developed to determine the relevant participants to contribute to the training process in each round. Hence, a good selection technique can significantly improve accuracy [7]. Several approaches leverage available client-side information. Some approaches use client-side metrics to select clients with high statistical utility, often determined by the significance of their updates [8]. Others target system-level characteristics and select clients based on computational capacity [9] or communication limitations [10]. However, these approaches treat all clients as equally representative and ignore differences in data and computational resources.

Data heterogeneity often arises from the non-uniform distribution of classes and features across the clients. Therefore, CS algorithms that consider only rich resource capabilities are often impractical in a healthcare environment. These approaches often remove resource-constrained clients from the training process and disregard the importance of domain diversity. Consequently, these methods repeatedly sample clients from the same domains, which limits the model's generalization to new or unseen data. Furthermore, medical data is typically scarce and highly heterogeneous. Consequently, an extreme selection strategy may exclude less favored clients and suppress important domain information, which may reduce the model's overall performance. To address these challenges, FL requires client selection mechanisms that handle both data heterogeneity and resource constraints. Therefore, a combined strategy is needed in the healthcare field to limit straggler effects and adapt to domain shift.

In this study, we propose CDCSF, a Domain-aware Client Selection framework, designed to ensure fair representation of all data domains while optimizing system efficiency. Our approach consists of two main components. First, inspired by prototype learning, we introduce cluster prototypes that capture domain characteristics. Each client is assigned to a corresponding domain cluster based on an EM clustering algorithm. This algorithm adjusts clustering based on clients' features, prototypes that optimized in each iteration. Second, we incorporate an efficiency and Fairness Client Selection that

jointly balances client training time and representativeness. This selection mechanism allows the server to detect stragglers proactively and penalize them without eliminating them. Additionally, clients with fewer selections are prioritized to improve fairness and ensure underrepresented clients contribute to training. Our approach balances statistical and system-level efficiency without compromising any client data distribution. CDCSF is designed for the healthcare domain, where the number of participating clients is typically small, fixed, and highly critical to the learning process. Clients often represent different domains, so domain shift becomes a significant challenge. Hence, in such settings, traditional client selection used in large-scale FL offers limited benefit. Excluding even a few clients is sensitive to the global model's generalizability. Nevertheless, client selection remains essential not for system optimization alone, but to ensure fair participation over time and preserve domain diversity. Neglect of any domain may introduce bias and harden the model generalization across heterogeneous data distributions.

Our algorithm is introduced to fill this gap in the literature. We believe that in constrained environments like healthcare, intelligent client selection remains critical not only for scale, but for representativeness, diversity, and robustness of the federated model. We highlight our main contributions as follows:

- We introduce a novel client selection strategy that accounts for domain heterogeneity across medical institutions. CDCSF ensures representation from diverse data distributions to enhance generalization.
- We propose a clustering framework based on the Expectation–Maximization (EM) algorithm that leverages local feature prototypes to partition clients into domain-homogeneous groups while preserving data privacy.
- The proposed client selection mechanism ensures a balanced trade-off between training efficiency and client participation. It promotes fairness by maintaining the involvement of underrepresented clients while moderately penalizing slower participants.
- CDCSF outperforms Power-of-Choice by 8% and FedAvg by 2% under domain shift, and reaches convergence two times faster than both baselines.

The study is organized as follows. Section I outlines the limitations of existing client selection approaches in healthcare settings under domain shift. Section II reviews related work and discusses its limitations. Section III presents our proposed method, CDCSF, a domain-aware client selection framework designed to address the limitations of current algorithms in the healthcare field under domain shift. We introduce two phases: First, a domain-aware clustering stage based on the EM algorithm that groups clients into different domain and second, a client selection stage that prioritizes fast clients through a reliability score based on training time, while also handling underrepresented clients through a fairness score. Section IV evaluates CDCSF on a medical benchmark dataset under domain shift and non-IID scenarios. Finally, Section V outlines the method limitations and future directions for advancing federated learning in healthcare.

II. Related Work

A. Domain Shift in Federated Learning

Domain shift arises from differences in acquisition equipment, annotation protocols, and patient demographics [11] introduce discrepancies in image quality and intensity profiles. These discrepancies can hinder model generalization. Recent methods have adapted Domain generalization techniques (DG) to FL to overcome domain shift. These approaches often require sharing intermediate feature representations [12] or domain-specific statistics [13] to align feature representations across clients [14]. Other approaches extend Domain-Adversarial Neural Networks (DANN) to the FL setting and employ discriminators to minimize feature discrepancies across clients [15]. However, many of these methods either require sharing their client data-related information or local representations. Although effective, these methods often raise privacy risks and increase communication overhead. To overcome privacy concerns, several recent works leverage features Prototype-based embedding to improve feature alignment. Features prototypes construct representative vectors by computing the mean vector for each class from each client [16]. The server aggregates these features to form global class prototypes. For instance, FedCCL [17] reduces client bias by aligning local and global class semantics. The server averages the local feature prototypes to construct a global prototype. This global prototype is then used to regularize local updates and capture finer intra-class structure. FedGFL [18] assigns adaptive attention scores to client updates through a nonlinear sigmoid function of their prototypical margins to guide model averaging. Similarly, FedProc [19] encourages each client's feature representations to match the global prototypes. However, these methods typically assume full client participation and synchronous communication, which is impractical in real-world settings with limited and heterogeneous resources. In real-world scenarios, typically only a small subset of clients is active per round, which can worsen the impact of data heterogeneity. In practice, client unavailability and computational heterogeneity cause delays which turn some clients into stragglers. As a result, addressing client selection becomes crucial when domain shift and system heterogeneity coexist. In this work, we leverage prototype learning as a privacy-preserving mechanism to construct a client clustering algorithm that identifies each client's domain throughout training. This domain-aware clustering serves as the foundation of our client selection framework to address data heterogeneity.

B. Client Clustering

Clustered federated learning (CFL) tackles data heterogeneity by grouping clients into clusters with higher internal similarity to improve model performance [20]. The core idea of CFL is to group participants with similar data distributions into the same cluster, and each group use its corresponding global model. This paradigm enables better adaptation to local data characteristics. FedAC [21] incorporates global knowledge into the clusters to maintain a balance between global and intra-cluster knowledge. The method splits the local model into distinct submodules and computes a distinct global model parameter for each cluster. It employs a lightweight approach to assess model similarity using dimensionality reduction. Additionally, it implements a

dynamic module to refine the number of clusters. Long et al. (2023) [22] use an Expectation-Maximization (EM) algorithm to compute the corresponding global models for each specific cluster. In the E-step, clients are iteratively assigned to clusters based on the proximity of their parameters to the previous iteration's cluster global models. In the M-step updates each cluster's global model by averaging the models of all clients assigned to that specific cluster. During local training, each client further adds a regularization term, specifically defined by the distance between its local model and its assigned cluster's global model. FlexCFL [23] addresses the computational challenges of high-dimensional similarity measures in traditional CFL. The method applies truncated Singular Value Decomposition (SVD) to reduce the dimensionality of the model updates. It then measures the similarity between each client's update and a small set of principal directions identified by SVD, which significantly reduces the high dimensionality.

C. Client Selection Algorithms

Client selection algorithms determine which clients participate in each training round. These methods have demonstrated their ability to improve model performance and strengthen their robustness. FedCS [24] aims to select the largest possible number of clients to complete the local training within a specific round deadline. The algorithm estimates each client's update and upload time based on the client's resource information. Ed-Fed [25] improve FL performance on mobile devices and predict the training time and battery drain of each client from resource information, such as RAM and CPU, which is communicated to the server before each round. The method uses this information and trains a neural network that predicts the training time and estimates battery consumption time. Ed-Fed selects clients that can run enough and finish in the same time and not exhaust their battery. [8] propose the POWER-OF-CHOICE algorithm that introduces a biased client selection. PoC method samples randomly a small candidate set of clients based on data proportions and then selects the m clients with the highest local training loss to participate in the training. However, these approaches don't take into consideration the data heterogeneity across clients in CFL. Authors in [26] propose a class-aware client selection that considers data heterogeneity to select a client. The algorithm groups clients based on their local label sets. Each client is assigned to a group where all members share the same set of labels. The method merges similar sets based on the Jaccard similarity of client label sets. In each communication round, the server selects the group with the least participation history to ensure fairness. However, in the healthcare field, clients exhibit significant domain shift due to feature skew. Hence, label-based grouping fails to capture the underlying data heterogeneity in such cases. Other works tackle the data and system heterogeneity jointly [27] Oort's computes its utility score based on statistical utility that is measured by local loss over selected samples and a system utility that is related to the expected training time. Oort penalize stragglers and guides the selection toward clients that maximize training progress per unit time. PyramidFL [28] introduces a fine-grained client selection framework. The statistical utility is estimated from historical training information such as local loss and update divergence. The system utility reflects training time, communication delay, and resource availability. The final utility score is computed as a function of statistical utility normalized by estimated round completion time, thereby prioritizing clients that contribute high-value updates efficiently. However, these approaches prioritize the fastest clients every time. While it is efficient but causes unfairness. Slower clients often remain unselected, so their data does not contribute to the global model, which can reduce its ability to generalize. To address this, authors in [29] introduce a long-term fairness constraint. Each client is guaranteed to be selected more than U times. This ensures that over the long training process, every client has a reasonable chance of being selected. The core idea is to introduce virtual queues for each client. These queues track the fairness constraint. If a client isn't selected often enough, its virtual queue length increases and the optimization framework will then prioritize selecting this client to prevent the queue from exploding. TiFL [30] addresses the challenge of fairness in CS. The method groups clients into tiers based on similar training times. In each round, the algorithm selects one tier to participate in training. To promote fairness, the method dynamically adjusts the tier selection probabilities based on training feedback. If a tier exhibits lower accuracy, its selection probability is increased. In this study, we group clients into domain clusters based on their feature prototypes. We further design a client selection strategy using both a reliability score, which penalizes persistent stragglers, and a fairness score, which identifies underrepresented clients and increases their selection frequency. A detailed comparison of prior client selection frameworks is presented in Table I. It summarizes the strengths and limitations of existing methods with respect to data heterogeneity and system heterogeneity in healthcare. These limitations highlight the need for CDCSF, which explicitly accounts for feature shift in federated healthcare environments.

TABLE I. A STRUCTURED COMPARISON OF CLIENT SELECTION MECHANISMS IN FEDERATED LEARNING

Method	Data Heterogeneity	System Heterogeneity	Key Limitation
FedCS [24]	It selects clients based on their computational capabilities, not their statistical contribution to the global model.	The method excludes clients that are too slow or resource-constrained.	The method offers no strategy to address domain or feature skew. As a result, it does not suit highly heterogeneous environments such as healthcare.
Ed-Fed [25]	Ed-Fed provides no mechanisms for handling data heterogeneity or domain shift.	Select clients that are able to finish local training without exhausting their resources.	Ed-Fed does not handle statistical heterogeneity or domain distinctions, which limits its use in healthcare FL practice. Furthermore, the method requires continuous reporting of device resource information, which introduces privacy and communication overhead.
PoC [8]	It samples a small candidate set of clients based on local data proportions	PoC does not handle system heterogeneity, as training-time	It does not consider the domain shift issue, fairness mechanisms, and tolerance for straggler clients.

	and then selects those with the highest local training loss.	differences and straggler effects are entirely ignored.	
Class-Aware Selection [26]	Addresses data heterogeneity by grouping clients according to their local label sets.	The method does not tackle the system heterogeneity issue.	Assumes that label distribution is the primary form of heterogeneity. It does not account for domain heterogeneity, which is critical in healthcare settings where clients differ more in feature distributions
Oort [27]	Oort jointly considers data heterogeneity by computing a utility score for each client. Its statistical utility is based on local sample loss and gradient divergence.	It computes a system utility which is related to estimated training time.	Slow or underrepresented clients are rarely selected. This limitation prevents Oort from fitting healthcare settings where every institution contributes essential data.
PyramidFL [28]	PyramidFL does not handle data heterogeneity.	The method favors fast, resource-rich clients and penalizes slower participants.	The method may exclude clients with valuable data.
TiFL [30]	TiFL does not addressed data heterogeneity.	It addresses system heterogeneity, where clients with similar training times group into the same tiers.	The method ignores feature skew, domain shift, and label imbalance across clients. Furthermore, slow tiers still produce noisy and low-quality updates.

III. METHODOLOGY

A. Problem Formulation

We consider a cross-silo FL setting with N clients (e.g., healthcare institutions) coordinated by a central server. System heterogeneity arises from differences in compute and bandwidth. Therefore, clients exhibit different training times. Accordingly, data are non-iid, and clients may suffer from a domain shift issue. In this section, we present CDCSF, a dynamic domain-aware client selection framework designed for the Federated learning (FL) setting.

Our framework comprises two main phases: (1) domain-aware client clustering and (2) an efficiency and fairness client selection. The overall system architecture is illustrated in Fig. 1. CDCSF performs domain-aware client clustering to identify groups of clients that share similar data distributions based on local feature prototypes. Clients send their local prototypes to the server in stage A. This phase explicitly identifies client domains to ensure that the overall data distribution is represented in every training round. Cluster assignments and center models are updated using an EM algorithm guided by cosine similarity (Stage B). Further, the server computes a composite score for each client by combining its reliability and fairness metrics. All clients across all clusters are then jointly ranked according to this global score. Client selection is performed at the cluster level: within each cluster, the server then selects the top-ranked clients according to the global score (Stage C). This ensures that client choices remain globally optimal while preserving balanced domain coverage. Finally, the server selects the client's participants and applies global aggregation (Stage D). Unlike static clustering approaches that fix group assignments before training, CDCSF continuously refines the cluster as the feature extractor evolves with rounds. As a result, the method continuously identifies and adjusts to evolving domain distributions during federated training. Moreover, CDCSF penalizes stragglers based on training time without excluding them entirely and introduces a fairness score that increases the selection frequency of underrepresented clients. Clients with limited resources may still hold informative data. Such data can significantly improve training when the number of available clients is small.

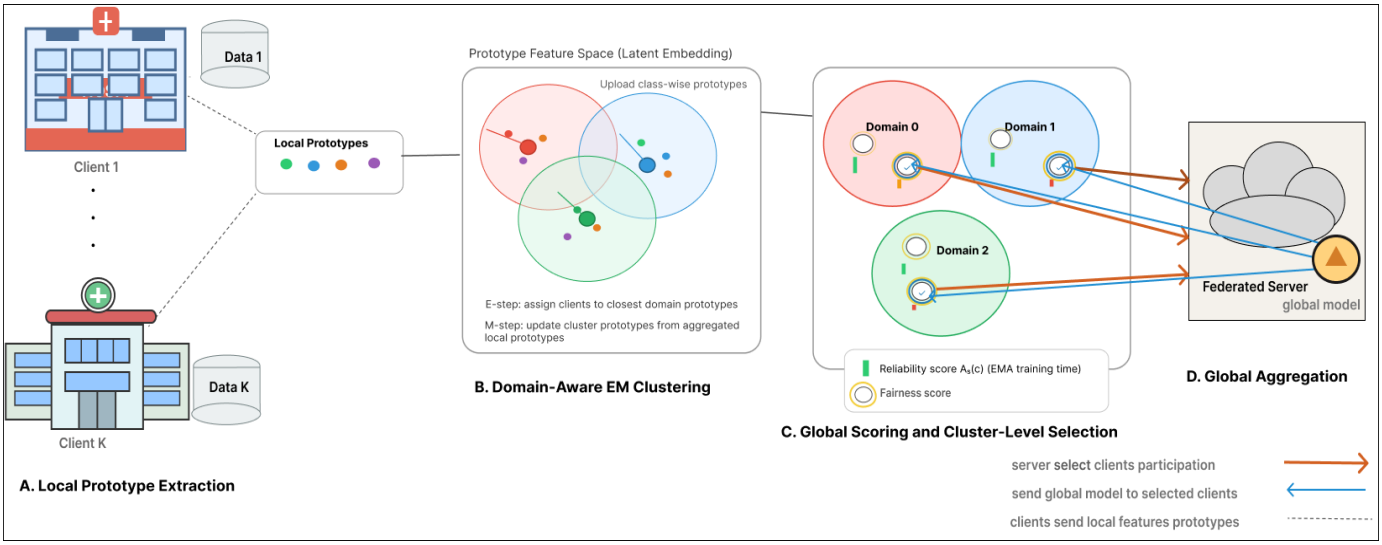

Fig. 1. Overview of the proposed CDCSF framework, where clients submit local feature prototypes, EM clustering identifies domain-aware groups, and the server applies client selection to choose participants for global aggregation.

Our objective is to learn function $f_\theta: X \to Y$, parameterized by θ, that maps input data from the input space X to their corresponding outputs in the label space Y. We consider a federated setting with m clients. Each client $i \in \{1, \ldots, m\}$ trains

a local model on its local dataset $D_i = \{x_i, y_i\}_{j=1}^{N_i}$, where N_i is the number of labeled samples on client i. We decompose the local model into two components: a feature extractorf: $X \rightarrow Z$ that maps input x to a d-dimensional representation$z = f_\theta(x) \in R^d$, and a classifier h: $Z \rightarrow R^{|Y|}$, where Y denotes the set of labels. Our goal is to learn a generalizable global model that performs well across non-iid client domains.

B. Domain-Aware Client Clustering

Our domain-aware clustering method groups clients based on their data distributions. Each cluster corresponds to a different underlying data domain. This task is challenging without direct access to raw data, since data is distributed across multiple clients. To address this limitation, we use feature prototypes to capture abstract domain-specific information [31]. We leverage local prototypes to effectively group clients into clusters that reflect shared domain characteristics. For each client i and class j, we define a feature prototype P_i^j as the mean of the embedding vectors for all samples labeled with class j. Let $S_j^i = \{n \epsilon \{1, \ldots, N_i\} | y_{i,n} = j\}$ denotes the index set of local samples on client i with label j. The prototype P_i^j is given as in Eq. (1):

$$P_i^j = \frac{1}{|S_j^i|} \sum_{(n) \in S_j^i} f_i(x_{i,n}) \quad (1)$$

where, f_i denotes the local feature extractor of client i. All clients share the same architecture but have client specific parameters during local training.

Due to privacy constraints, raw data and feature representations cannot be shared directly [32]. Instead, local prototypes are privacy preserving act as summaries of the local data distributions. These vectors are the average of an embedding vector. Prototypes offer an efficient mechanism to infer latent domain similarities across clients from class-conditional feature structure, which is captured by each specific client. As a result, they are effective for clustering clients that share similar underlying data characteristics.

We introduce a clustering procedure based on the Expectation-Maximization (EM) algorithm. The algorithm partitions clients into domain-homogeneous groups through two iterative steps. The expectation step (E-step), which assigns each client to the cluster whose prototype is most similar to the cluster-level prototypes. The maximization step (M-step), which updates the global cluster prototypes by aggregating the local feature representations of the clients assigned to each cluster. Our approach leverages class-specific feature prototypes to capture the statistical characteristics of each client's local data distribution. The EM procedure updates cluster assignments and prototypes throughout training. As a result, it supports adaptive grouping of clients and mitigates distributional discrepancies across institutions. These prototypes serve as a compact representation of domain-specific information and guide the clustering process.

In the E-step, each client i is assigned to a single cluster k via hard assignment. For each client, we compute the aggregated cosine distance between its local class-wise prototypes $\{P_i^j\}_{j=1}^C$ and the global cluster-level prototype $\{\tilde{P}_{j,l}\}_{j=1}^C$, for $l \epsilon \{1, \ldots, K\}$ and $C = |Y|$. Let $d(.,.)$ denotes the cosine distance. Client i is assigned to cluster k that minimizes this distance. It is determined in Eq. (2):

$$s_i^k = \begin{cases} 1, \text{if } k = \arg\min_l \sum_{j=1}^C d(P_i^j, \tilde{P}_{j,l}) \\ 0, \text{Otherwise} \end{cases} \quad (2)$$

where, $s_i^k \in \{0,1\}$.

Algorithm 1: Domain-aware client clustering

Initialize: A: set of all clients, T = total training rounds, α = EMA, η = learning rate, K: initial number of clusters, E: re-cluster interval, C_k: client clusters, D_i : local dataset of client i, E: number of local epochs, B: batch size, f_θ:feature extractor , h_ρ : classifier

Aggregator: initialize weights for all clients W_0

Client side

Receive global model from server W_0

For each client i in parallel do

  For each epoch = 1…E do

    For each batch in B do

      $z_b = f_\theta(x\_b)$

      $\hat{y} = h_\rho(z_b)$

      $L_i = \frac{1}{|B|} \sum \log\left(\frac{e^{\hat{y}_{b,y_b}}}{\sum_j e^{\hat{y}_{b,j}}}\right)$

      $w_t \leftarrow W_t - \eta \nabla L_i$

      For each class j in D_i do

        $S_{i,j} = \{(x,y) \in D_i \mid y = j\}$

        Compute P_i^j with Eq (1)

        End

    End

  End

Send local parameter w_i local prototypes P_i^j and $|S_{i,j}|$ to server

Server side

For each client c in A in parallel do

$$W_{t+1} \leftarrow \sum_{i=1}^{N} w_t$$

Server collects latest normalized prototypes of each client c that reported P_i^j.

Request prototypes: p[c] ← GetPrototypes(c)

E-step: assign client c to cluster k using (2): client_assignments[c]=k

M-step: Update cluster prototypes using (3)

return client_assignments to resource-aware algorithm

return global parameter W_t to each client

End

The cluster index assigned to client i is: $k = \arg\min_l \sum_{j=1}^C d(P_i^j, \tilde{P}_{j,K})$.

Once all clients are assigned, the M-step then updates the global prototype $\tilde{P}_{j,K}$. It is computed as a weighted average of local prototypes from all assigned clients to that cluster as in Eq. (3):

$$\tilde{P}_{j,K} = \frac{\sum_{i=1}^{m} s_i^k |N_{j,i}| \, P_i^j}{\sum_{i=1}^{m} s_i^k \, |N_{j,i}|} \quad (3)$$

Where $|N_{j,i}|$ is the number of samples for class j on client i, and m is the total number of clients in the cluster k.

We use cosine similarity to compare local and global cluster prototypes. This choice is consistent with prior prototype-based FL methods [19]. Cosine distance emphasizes directional similarity and reduces sensitivity to magnitude differences. This property makes it more suitable for semantic similarity between class-conditional embeddings. Qiao et al. (2025) [17] have demonstrated that cosine distance outperforms Manhattan and Euclidean distances for prototype features representation.

Our domain-aware client clustering methodology employs the EM algorithm to iteratively partition clients into domain-homogeneous groups. The clustering is based on class-specific feature prototypes, as illustrated in Algorithm 1. The server first sends the initialized parameters to all clients (line 3). Each client then performs local training to compute model parameters (lines 5-10). After local model optimization, each client uses its feature extractor on training data to generate local feature prototypes P_i^j (line 13). These local embeddings are then transmitted to the server for the t-th communication round. We assume that the set of responding clients is denoted by S_t. The server updates the global model parameters by averaging the received $\{\theta_{t+1}^k\}_{k \in S_t}$, following the standard aggregation rule in FedAvg. Moreover, for the global prototype update, we introduce a new strategy that applies an EM algorithm to cluster clients into domain-homogeneous groups. In E-step (line 21), each client is assigned to the cluster that minimizes the aggregated cosine distance between their local prototypes and global cluster-level prototypes across all classes (line 21). In the M-step (line 23), the global prototypes are updated as weighted averages of local prototypes from clients assigned to each cluster. The weights are proportional to the number of samples per class. The role of EM is to assign each client to its corresponding cluster (line 24) and return these groups to the second phase of client selection. Finally, the server broadcasts the global parameter to each client (line 26) for the next local training iteration.

The EM algorithm is sensitive to two key hyperparameters: the initial number of clusters K and the re-clustering interval E. Regarding K, our framework exhibits an important asymmetric sensitivity. When K is initialized larger than the true number of underlying domains. The dynamic nature of our EM clustering provides a natural self-correcting mechanism. As training progresses and feature prototypes become increasingly discriminative, clients that belong to the same true domain will produce semantically similar prototypes. Consequently, the cosine distance between their local prototypes and a shared cluster-level prototype will decrease. As a result, these clients converge toward the same cluster assignment. Over time, the effective number of active clusters reduces toward the true domain count. This dynamic clustering is a direct consequence of using learned, progressive prototypes rather than fixed statistics. In addition, the re-clustering interval E controls how frequently cluster assignments are updated. Smaller values of E allow clusters to adapt more quickly to changes in the feature space. In contrast, larger values reduce computational and communication overhead but may delay the correction of incorrect cluster assignments.

C. An Efficiency and Fairness Global Client Selection

We propose a time-optimized and fairness global algorithm that dynamically selects the best set of clients from all clusters based on their real-time training and their historical participation in the training process. The approach evaluates participants according to their consistent engagement throughout the training. This involves adding a reliability score, which tracks clients' training times to distinguish between less responsive and more efficient clients. In addition, a fairness score is incorporated to ensure that clients with historically low participation are selected more frequently.

1) Reliability score: FL performance is constrained by the varying time each client needs to complete local training or transmit model updates to the server. These slowdowns caused by clients with poor network connectivity or low computational power are known as the straggler effect. The random client selection used in FedAvg effectively works only when no stragglers are present. Favoring resource-rich clients prevents slower clients from often contributing to the training. Hence, reduce global model's generalization leads to a loss of fairness in the learning process. To address this problem, [24] presents a client selection method that collects the maximum number of updates within a predefined deadline. Similarly, [30] proposes a tier-based approach that chooses clients with similar training durations per round. Nevertheless, such strategies may remove clients who possess statistically valuable data from the training process. This score enables the server to identify persistently slow clients without removing them entirely from the federation. Clients with high latency receive a proportional penalty through the reliability score. This penalty allows the server to prioritize faster participants when efficiency is required. The score preserves opportunities for slower but important clients to contribute to the training process. We avoid hard exclusion and rely instead on adaptive penalization. Hence, our reliability mechanism maintains system efficiency. This is an essential requirement in healthcare FL, where the loss of a single institution can eliminate an entire domain of patient demographics or imaging modalities.

In this setup, the available time of client c is defined as the duration required to complete the server's training request, which is measured when the client requests the aggregated weights from the server and starts its training process. Client computational heterogeneity obliges the server to wait for all clients' responses before performing global aggregation. This behavior increases idle time for faster clients. Our approach records the training time for all clients at each round. To characterize a client's responsiveness, we define $t_{avail,c}(i)$, as the training time of client c in a given round i. Since a client may perform efficiently in one round and more slowly in another, its performance is evaluated across all previous rounds rather than

a single observation. We propose a responsiveness $T_c(i)$ based on the Exponential Moving Average (EMA) of the training time for every client. $T_c(i)$ integrate recent and past performance information. It prioritizes clients with consistent reliability performance. We define $T_c(i)$ as in Eq. (4):

$$T_c(i) = \alpha t_{avail,c}(i) + (1-\alpha)T_c(i-1) \quad (4)$$

Here, $\alpha \epsilon [0,1]$ determines the trade-off between recent and older observations.

Eq. (4) focuses on consistent speed metrics and prevents fast clients from being incorrectly labeled as stragglers due to occasional delays. Consequently, it yields a more reliable and stable measure of client speed.

In synchronous FL frameworks, most methods use a threshold to decide which clients can participate in the training process. This mechanism exclude client permanently based only on the client's current training time. These thresholds are often predefined before training begins. Without any other knowledge of client performance behavior, they typically require extensive manual tuning. This limitation becomes critical in practical scenarios where new clients may join the training process with unknown computational capabilities, or when a majority of clients exhibit slower processing speeds. In such cases, indiscriminately penalizing or excluding all slow clients is both inefficient and counterproductive. Our method addresses this issue by evaluating each client's performance relative to the historical training times of all participants, enabling more context-aware and adaptive decision-making. This design is particularly important in domains such as healthcare federated learning, where each client possesses unique and irreplaceable local data distributions that are essential for building robust and generalizable global models. The reliability score is computed as in Eq. (5):

$$A_s = \frac{T_{max}}{(T_c + \beta T_{max})} \quad (5)$$

Where $T_{max} = \frac{1}{N}\sum_{c=1}^{N} T_c(i)$ and $\beta \geq 1$ controls penalty strength.

This equation is sound and effective. It penalizes clients whose average training time (T_c) exceeds the historical average of all clients (T_{max}). The use of an adaptive threshold (T_{max}) makes it robust to varying system conditions.

2) Fairness score: To achieve good training time, we reduced the selection of slower clients across rounds using the reliability score. However, this improvement in training speed might compromise the model's generalization. Repeatedly selecting only faster clients can bias the global model to overfit towards the data of often-presented institutions. This behavior might underrepresent important domains and limit the model's robustness. Therefore, it is important to give underrepresented data more chances to contribute to training. This is essential, as no institution should be permanently excluded from collaborative learning.

Our method ensures that every client remains eligible to contribute. To achieve this, we propose a new fairness score that increases the selection frequency of clients with historically low participation. We define f_s which quantifies a client's historical underrepresentation in the training process. The fairness score for a client c is a continuous, decaying function based on their selection ratio R_c. We define R_c in Eq. (6) to measure the underrepresentation of a client c based on a client's selection:

$$R_c = \frac{v}{\frac{T}{n} * K} \quad (6)$$

Algorithm 2: Adaptive participation for client selection

```
Inputs  T_warmup: number of warm up iteration, stage: the stage
        Of selection (domain or warmup), participated_clients:
        participated_clients , virtual_cluster_enabled:
        Boolean flag for virtual cluster usage.
1       participated_clients ← ∅
2       selection_counts ← {}
3       virtual_cluster_enabled=TRUE
5       For t in T-1 do
6           For each client in clients do in parallel:
7               Clusters ← {}
8               perform_clustering ← FALSE
9               If t ≤ T_warmup then
10              stage ← WARMUP
11              use_clustering ← FALSE
12              Else
13              stage ← DOMAIN_AWARE
14              use_clustering ← TRUE
15              If use_clustering then
16              perform_clustering ← TRUE
17              Else
18              Perform_clustering ← FALSE
19              If stage = WARMUP Or client_assignments
                is empty then:
20              Clusters[0] ← C_participated ∪ c
21              Else
22              For each client c
                ∈C_participated do
23                  If c in client_assignments then:
24                  k ← client_assignments[c]
25                  Clusters[k].append(c)
26                  Else
27                      Clusters[0].append(c)
28              If c not in participated_clients And
                virtual_cluster_enabled then:
29              k_virtual ← K
30              Clusters[k_virtual] ← C
```

where v represents the cumulative number of times client c has been selected for participation, T denotes the total number of completed training rounds in iteration i, N represents the total number of available clients, and K is the number of participants

in each round. This ratio quantifies whether a client has been selected proportionally to the ideal uniform distribution, where each client should participate in $\frac{TK}{n}$ rounds on average. The fairness score is then calculated using the inverse relationship defined in Eq. (7), which assigns higher fairness weights to underrepresented clients ($R_c < 1$) and lower weights to overrepresented clients ($R_c > 1$):

$$f_s = \frac{1}{1+R_c} \tag{7}$$

This approach guarantees that even clients who have been selected more than the ideal average still retain a small, non-zero fairness weight. This dynamic promotes balanced and fair participation during the entire learning process.

3) Global score: Given both data and resource heterogeneity, traditional client selection methods are largely insufficient for healthcare applications. Resource-aware strategies often exclude many clients from global aggregation because of their limited computational or communication capabilities. This preference systematically removes a larger number of relevant clients. In non-IID settings, clients may hold imbalanced data quantities or represent distinct domains, so excluding any institution can degrade the performance and robustness of the global model. To address this limitation, we present a global scoring mechanism in our clustering framework to achieve an adaptive balance between computational efficiency and participation fairness. The server computes a global score for each client and ranks all clients across clusters. It then identifies the top-k clients within each cluster and selects them for the next training iteration. This design enables the system to dynamically adjust selection frequencies across rounds in response to changing computational and statistical conditions. The global score uses historical training time patterns to identify stragglers relative to their peers of other clients (4–5) and adjusts their influence in a manner that avoids permanent penalization. Our method defines the global score as in Eq. (8):

$$s_c = \alpha_1 A_s + \alpha_2 f_s \tag{8}$$

α_1 and α_2: a balance between fairness and reliability of clients, and it can be tuned by users.

This equation enables explicit tuning of selection priorities. During early training, a higher α_1 can prioritize fast and reliable clients to stabilize the initial model. As training progresses, increasing α_2: ensures that underrepresented clients receive more opportunities. This improves domain coverage and enhances the model's generalizability. In healthcare-oriented federated learning, this approach addresses two fundamental challenges. First, it preserves institutional equity while maintaining the unique data contributions of each participating hospital or medical center. Second, it ensures that resource-constrained institutions are not systematically excluded from collaboration, which helps retain the diversity and generalizability required for developing robust medical AI models.

As illustrated in Algorithm 2, the client selection mechanism proceeds through several phases designed to balance training efficiency, domain diversity, and participation fairness. The clustering algorithms depend on learned feature representations. However, in early rounds, randomly initialized models produce unreliable embeddings that do not reflect true data distributions. Therefore, we propose a warm-up period to allow the model to learn domain-specific features. Our approach clusters clients using learned feature representations rather than raw statistics. Domain heterogeneity is determined by the internal representations that models acquire rather than by the raw data itself. For this reason, most existing methods aim to learn domain-invariant feature representations to mitigate this issue [25]. During the warm-up phase, all clients are treated as a single cluster, and the global selection score operates over the entire client set. This phase stabilizes training and prioritizes fast, consistent clients to build a strong initial global model.

Algorithm 3: An efficient and fair global client selection

Inputs Clusters: set of clients in a cluster k, $t[c]$: time availability score of client c, T_{avg} : average training time of all clients, $t_{avail,c}$: training time of client c, T: total training rounds , α : EMA smoothing factor, β = reliability penalty parameters, Q: number of selected clients in each cluster, selection_counts: tracks selection frequency per client,t_hist: training history times of selected clients

1 For each cluster k in Clusters:

2 candidates ← Clusters[k]

3 For each client c in Clusters[k] do:

4 If c *not in* t_hist then:

5 t[c]=$t_{avail,c}$

6 Else:

7 $t[c] = \alpha t\_avail + (1-\alpha)$ t[c]

8 $T_{max} = \frac{1}{N}\sum_{c=1}^{N} t[c]$

9 End

10 *compute* A_s with Equation (3)

11 If

12 $f_s[c] \leftarrow T / (T + selection\_counts[c] \times n)$

13 Else

14 $f_s[c] = 1.0$

End

15 Compute global score score with (8)

16 participated_clients ← participated_clients ∪ {c}

17 selection_counts[c] ← selection_counts[c] + 1

18 End

For cluster in candidates do

19	candidates_sorted ← Sort (candidates, key= score, reverse=True)
20	selected_k ← candidates_sorted[0:Q]
21	Clusters_tos[k]= selected_k
22	End
23	Return Clusters_top

In later phases, we increase fairness to ensure that all clients can contribute. After the warm-up period, the algorithm enters the domain-aware phase. It first records baseline participation patterns (lines 8–10), then performs clustering to identify groups reflecting underlying data domains (lines 12–16). After the clustering phase, clients are organized into groups for selection (lines 19-27). Moreover, A challenge arises due to partial participation in federated learning. FedAvg aggregates updates only from selected clients, while non-participating clients do not contribute to model updates. As a result, the prototype collection is biased toward previously selected clients, and the EM clustering algorithm subsequently operates on a limited set of prototypes. This bias may cause cluster assignments to reflect historical selection patterns rather than true domain characteristics of all clients. To address this limitation, we introduce a virtual cluster appended to the EM-derived clusters. This additional cluster represents clients that have not yet participated in training, thereby preventing systematic exclusion and preserving domain diversity. we initialize the fairness score to its maximum value ($f_s = 1.0$) for clients that belong to the virtual cluster (lines 28-30). This initialization ensures that the global selection score considers these clients and ranks them alongside active participants. As a result, previously inactive clients receive an opportunity to contribute to training and become eligible for the domain clustering algorithm in the next rounds.

Algorithm 3 outlines the client selection procedure executed by the server during iteration t. The selection mechanism samples clients from both the EM algorithm and the virtual cluster (line1). This strategy allows the clustering process to access a broader range of client information, which helps preserve representativeness and reduce selection-induced bias. At each iteration t, participating clients report their training time $t_{avail,c}(i)$ to the server. Each client c updates its historical training time $T_c(i)$ using Eq. (4) (line 7). Once all client reports are received, the server calculates the average training time T_{max} (line 8). After completing clustering assignments using the EM algorithm for all clients, the server computes two scores, respectively: first, a reliability score (line 10) that evaluates temporal efficiency and a fairness score that measures participation equity (line 12). For previously selected clients, the fairness score is updated using Eq (8). For new clients, the fairness score is initialized to the maximum value ($f_s = 1.0$) to guarantee initial participation (lines 13). Finally, the server integrates these components to compute the global selection score s_c (line 15). This score provides a global ranking of clients, independent of their cluster assignment. The server enforces cluster-level selection. Within each cluster, clients are sorted by their global score, and the top-k clients are then selected. Thus, while the ranking is global, the final selection is performed within each cluster. This design balances fairness and efficiency globally while ensuring domain diversity. First, it reduces the chance that a single domain dominates selection simply because it contains many high-scoring clients. Second, it prevents the exclusion of minority domains and provides broader representation in training. Third, the approach maintains sensitivity to system heterogeneity, as slow clients are penalized regardless of their domain. This aligns with our motivation that GDCSF balances straggler mitigation, domain diversity, and fairness toward underrepresented clients.

IV. EXPERIMENT

A. Implementation Setup

We evaluate our method on the PathMNIST benchmark for multi-class medical image classification [33]. In a typical federated learning (FL) setting, we compare CDCSF with two standard FL baselines, FedAvg and Power-of-Choice (PoC), as well as with a centralized learning model. The evaluation is conducted under conditions of domain shift and the presence of straggling clients, which reflect realistic FL deployment scenarios. To emulate domain shift across clients, we partition the system into four domains representing distinct imaging conditions. Specifically, we split the PathMNIST training set into three separate domains, while the test set is assigned to an additional client, which serves as a fourth independent domain. Domain 0 corresponds to the original domain of the dataset. Domain 1 simulates mid-range equipment commonly found in resource-constrained settings and introduces a modest change in brightness (between −0.2 and +15%), reduced resolution, and contrast variations in the range [0.6, 1.4]. Domain 2 represents degraded imaging conditions, modeling more challenging real-world scenarios via increased brightness (+0.3), additive Gaussian noise ($\sigma = 0.12$), and contrast variations in [0.5, 1.5]. To additionally examine model behavior under system heterogeneity, we designate a subset of clients as stragglers throughout the entire training process. Whenever a straggler is selected for participation, we impose an additional delay sampled uniformly at random from the interval [10, 20] seconds for each local training phase. Although the added delay does not represent actual hospital latencies, it still produces the same relative effect. Fast clients proceed at normal speed, whereas stragglers require two to three times longer to complete their local updates. This controlled slowdown reflects the variations in hardware performance, workload, and network connectivity found across healthcare institutions, where some sites consistently operate more slowly than others. All experiments use a ResNet-18 backbone as the core feature extractor to ensure a fair comparison between all the methods.

The ResNet architecture was adapted for the PathMNIST dataset, which consists of 28×28 RGB images. To accommodate the smaller input size, the initial convolutional layer was modified to use a 3×3 kernel with stride 1 and 32 initial feature maps, followed by four residual blocks with progressively increasing channel dimensions (32, 64, 128, 256). Each block comprises two convolutional layers, batch normalization, and ReLU activation, followed by an adaptive average pooling layer and a fully connected classification head. The CDCSF model integrates this ResNet backbone within a federated learning framework, where the feature extractor is complemented by a projection module for prototype extraction and clustering. This

module consists of two linear layers with ReLU activation, reducing the feature dimensionality to 128 while preserving discriminative representations through normalization. The classification head is implemented as a single fully connected layer mapping the extracted features to the nine target classes in PathMNIST. Federated training was implemented with the Flower framework [34]. We conduct experiments with K = 10 and K = 20 clients. For K = 10, we sample 4 clients per round, and for K = 20, we sample 6 clients per round. Each selected client performs 3 local training epochs. Training runs for 200 global rounds until convergence. To ensure a fair comparison, all federated learning methods use the same hyperparameters, batch size of 32, and a learning rate of 0.01. All experiments are executed on two NVIDIA T4 GPUs.

B. Results and Discussions

We evaluate the proposed CDCSF framework under two key challenges in federated learning: (1) data heterogeneity and (2) system heterogeneity. We compare CDCSF with two FL baselines: FedAvg and Power-of-Choice (POW) and centralized learning. Table II reports the average test accuracy under domain shift and system heterogeneity. CDCSF achieves the highest final accuracy among all methods and surpasses FedAvg by 1.7 %. It also converges approximately two times faster. CDCSF shows strong effectiveness under a distribution shift. First, the domain-aware clustering groups clients according to their latent data distributions. This ensures that each round selects clients representing diverse domains within the global data distribution rather than sampling from randomly biased domain sources. Second, during the client selection, the global score assigns a higher historically underrepresented. This reduces early-round instability, especially under severe domain shift. CDCSF consistently surpasses Power-of-Choice (POC) with an improvement of more than 8% in final accuracy. This improvement is expected because POC evaluates clients only through their performances, which assigns higher weights to clients with higher loss and skews the optimization toward dominant domains. However, in domain-shifted medical data, this type of selection intensifies sampling bias, reduces the participation of minority domains, and increases the influence of the overrepresented clients. In contrast, CDCSF preserves domain diversity through prototype-based clustering and gives priority to reliable clients in early rounds. As a result, CDCSF selects a more informative and balanced client subset in each round. The results show that CDCSF fits heterogeneous medical settings more effectively, where strong performance requires both computational efficiency and balanced domain representation.

TABLE II. ACCURACY IMPROVEMENT OF CDCSF COMPARED WITH FEDAVG, POC, AND CENTRALIZED LEARNING UNDER DOMAIN SHIFT

Algorithm	Accuracy
FedAvg	82.02%
PoC	66.37%
CDCSF	83.78%
Centralized Learning	86.23%

To demonstrate the benefits of our framework under domain shift, we present the validation accuracy of all FL methods in Fig. 2. For clarity, note that the validation accuracy reported for each method is computed as the mean validation accuracy of all clients. CDCSF achieves faster convergence in early rounds compared to FedAvg and Power-of-Choice (POC). Our method reaches 80% accuracy within 75 rounds, two times faster compared to FedAvg, while POC fails to attain this accuracy. During the early iterations, when the model begins to converge, CDCSF selects clients from diverse domains. This ensures that heterogeneous data sources contribute to the global model. Hence improves generalization performance. In addition, the algorithm initially prioritizes fast and reliable clients to accelerate convergence. As training progresses, we increase the weight of the fairness score to enable slower or previously underrepresented clients to join the training process. Further, we highlight the effect of distribution shift on the compared client selection frameworks. Domain heterogeneity generally reduces performance, but its impact on FedAvg remains comparatively modest, as shown in Table II and Fig. 2. This behavior aligns with our motivation because FedAvg selects clients uniformly at random, which ensures that all clients from different domains participate with uniform equal frequency. Under domain shift, such uniform selection benefits the global model. As a result, exposure to all domains enables the model to learn efficient domain-invariant representations. Consequently, we also observe that FedAvg converges faster than (POC), which suffers from a selection bias that oversamples certain domains while underrepresenting others. This bias degrades the model's performance by repeatedly prioritizing clients with temporarily higher losses. These observations show that distribution shift harms client-selection algorithms that rely primarily on resource availability or short-term statistical performance.

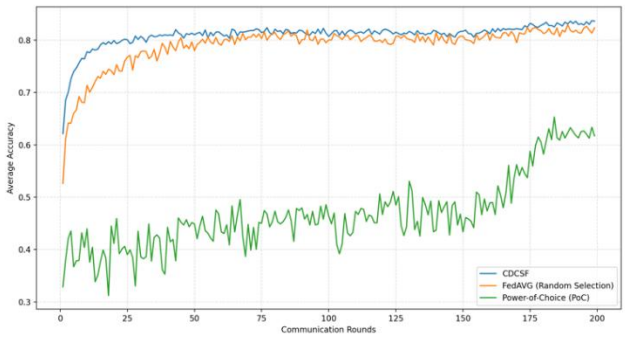

Fig. 2. Average validation accuracy of CFDSC, FedAvg and PoC across communication rounds.

As illustrated in Fig. 3, FedAvg selects clients at similar frequencies, but CDCSF produces a more balanced and adaptive pattern that reflects both efficiency and domain diversity. Although uniform sampling guarantees equal selection frequency in FedAvg, this property is insufficient in healthcare FL, where clients suffer from a domain shift issue. This sampling neglects the existing domain imbalances, enforcing equal participation frequency across clients without addressing underlying distribution disparities. CDCSF addresses this limitation through a domain-aware clustering mechanism that groups clients using privacy-preserving feature prototypes. The server applies an EM algorithm to identify heterogeneous clinical domains by clustering clients according to their underlying data distributions. As a result, even clinically

important domains with only a few clients are placed into their own clusters rather than being merged with the majority of domains. Moreover, the incorporation of EMA reliability scoring mitigates the impact of stragglers without permanently excluding slower institutions, while the fairness score ensures equitable participation for historically underrepresented clients. The global scoring mechanism then selects the top-k clients from each cluster and ensures consistent representation of minority domains throughout training. These components form a coherent and robust improvement over random selection. The method increases representativeness, reduces domain bias, and supports more efficient convergence in federated healthcare settings.

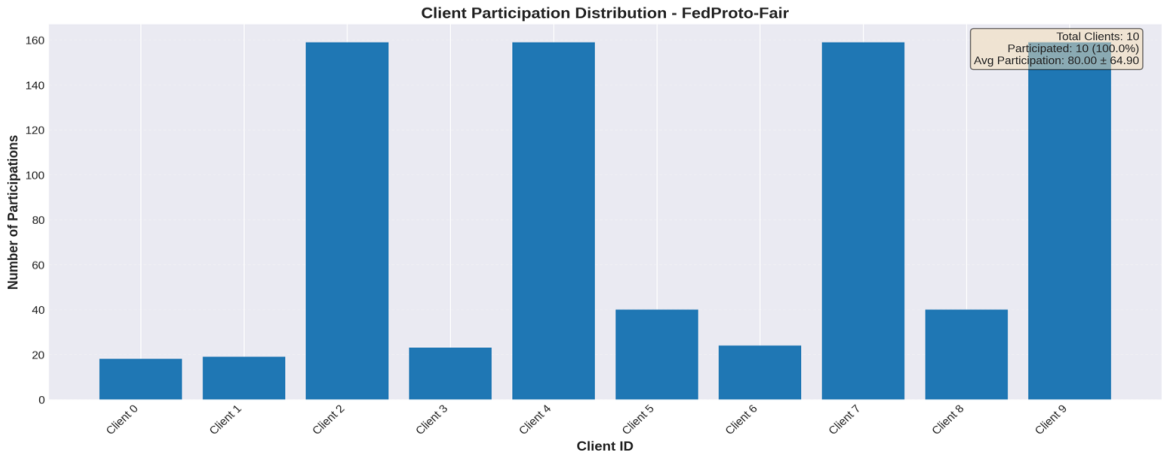

Fig. 3. Comparison of client participation frequencies across methods.

To model system heterogeneity, we define stragglers as clients who incur additional delay during local training. In our simulation, we assign this role to clients 0 and 1, while all other clients operate normally.

The server computes an exponential moving average (EMA) of each client's training time and computes a global reference value T_{max} at every round, obtained as the mean EMA across all clients. Fig. 4 displays each client's EMA estimate T_c together with the global reference T_{max}.

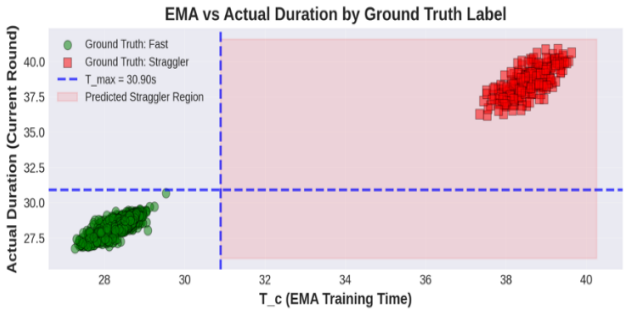

Fig. 4. The detection of stragglers' performance using EMA reliability scoring.

The straggler clients, shown in red, fall in the 37–40 second range, whereas fast clients, shown in green, cluster around 28–30 seconds. The two groups remain well separated. The vertical dashed line at $T_{max} = 30.9$ Seconds highlights this separation. The shaded region corresponds to the decision boundary of our reliability score. A client falls inside this region when its score $s = 1 - A_s$ exceeds the threshold θ. We use this score to predict whether a client is a straggler or not. The visualization confirms that our detection rule remains stable across all 200 rounds, despite the random injection of extra delay.

To further examine how the reliability score prioritizes fast clients, we train the model with k=20 clients and visualize the relationship between each client's reliability score and its EMA training time. Fig. 5 illustrates how training duration affects the reliability metric. Clients with larger T_c values (e.g., Clients 0 and 1) appear in the lower right region of the plot and receive lower reliability scores, which reflects the inverse relationship between training time and reliability. In contrast, faster clients form a compact cluster in the upper left region, indicating consistently higher reliability. This visualization confirms that the scoring mechanism effectively separates slow and fast clients and demonstrates that faster clients get higher prioritization weights. Importantly, the purpose of the reliability metric is not to penalize or exclude stragglers. Instead, it assigns them appropriately reduced weights while preserving their participation. Their inclusion is reinforced through the fairness component, which ensures equitable representation so that the global model still benefits from their unique data characteristics.

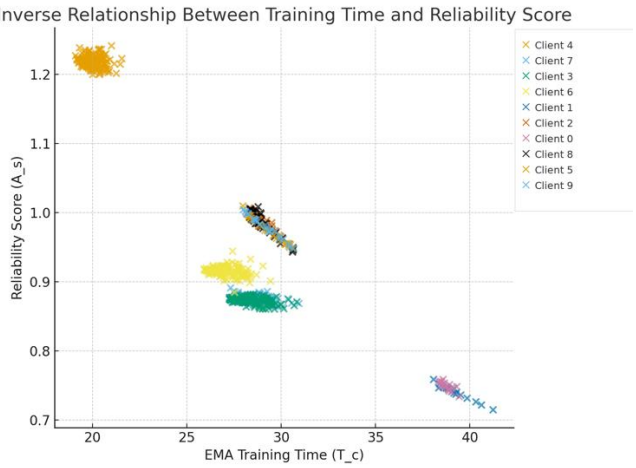

Fig. 5. Relationship between EMA training time and reliability score.

To further evaluate the impact of data heterogeneity, we partition the dataset across clients using a Dirichlet distribution with concentration parameter α=0.5, which creates a highly skewed non-IID distribution. Fig. 6 visualizes the distribution of each class across clients for all domains.

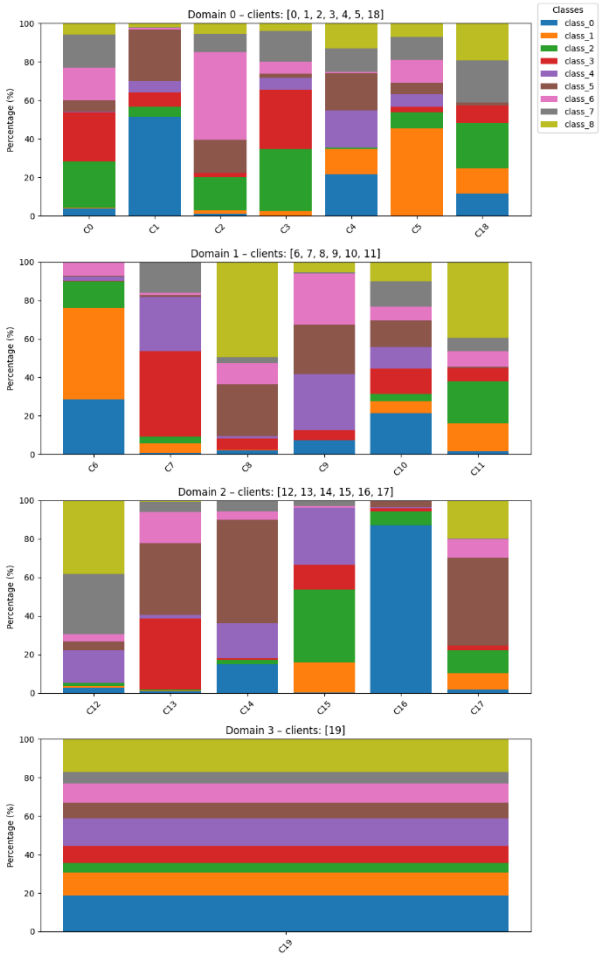

Fig. 6. Visualization of non-iid client distributions using Dirichlet sampling under domain shift when ($\alpha = 0.5$).

In addition, we apply a severe domain shift to each domain to increase the difficulty compared to our initial simulation. In this experiment, only 6 out of 20 clients are selected per round. In practical FL deployments, data heterogeneity is the norm. Therefore, this analysis allows us to evaluate how CDCSF simultaneously addresses both data and resource heterogeneity. In this setting, we test multiple scheduling policies from the CDCSF framework. Each policy selects clients based on global scores that combine reliability and fairness weights. Each configuration produces a distinct CDCSF graph. The reason for that is to show different trade-offs between efficiency and participation. We compare these policies with the FedAvg baseline. We refer to this algorithm as vanilla. As expected, the overall accuracy in Fig. 7 is lower compared to the results shown in Fig. 2. This behavior is expected because the data distribution in this scenario jointly combines feature skew and label imbalance, both of which are known to disrupt the performance of federated optimization. Our method does not attempt to correct the data distribution issue at its source. However, we observe that CDCSF (hybrid) shows a stable and consistent learning trajectory. In contrast, FedAvg demonstrates pronounced fluctuations characterized by inconsistency across rounds, which indicates its sensitivity to non-IID distribution. These results demonstrated the role of CDCSF in mitigating the effects of data and resource heterogeneity. This improves the convergence rate even when both conditions coexist. The results in Fig. 7 further illustrate the impact of our three selection strategies. CDCSF–fast prioritizes clients with high reliability scores and favors devices that have historically shown faster performance. CDCSF–equal assigns equal weights to fairness and reliability across all rounds. CDCSF– hybrid adopts a dynamic policy that emphasizes reliability in the early rounds and increases fairness weight in later rounds to incorporate underrepresented clients more effectively. This adaptive behavior allows CDCSF–hybrid to maintain efficient training while progressively enhancing participation diversity. During the first 20 rounds, which serve as the warm-up phase, we do not employ the EM domain clustering. All methods behave similarly and show nearly identical performance. As expected, this confirms that, in the absence of clustering, the client selection process behaves like random sampling. In this early stage, training time does not provide meaningful information for distinguishing reliable from non-reliable clients, so the global selection mechanism does not contribute to performance improvement. Once the warm-up phase finishes and EM clustering becomes active, the performance of the methods begins to diverge. We observe that CDCSF–hybrid achieves the best convergence and outperforms all other policies. Its design prioritizes reliable clients in the early iterations, which stabilizes learning and refines the local feature-prototype representations that guide domain assignment in the clustering. As training progresses, we increase the fairness weight so that unrepresented clients are then incorporated in domain coverage without disrupting optimization. Introducing fairness in the mid-stage is essential because it allows underrepresented clients to enter the global score ranking and ensures their participation from the mid to late rounds. This gives these clients sufficient opportunity to contribute their updates and influence the global model. This adaptive balance between early reliability and later fairness enables CDCSF–hybrid to achieve both stable convergence and improved generalization across heterogeneous domains. The behavior observed in the fast and equal policies further supports this interpretation. The fast policy outperforms the equal policy because, under imbalanced data distributions, FL algorithms naturally benefit from consistently available participants who contribute updates regularly. Similarly, FedAvg, which relies on random selection, outperformed the equal policy, which confirms that participation consistency plays a central role in FL performance. However, relying on consistent clients leads to biased learning. CDCSF–hybrid overcomes this limitation by ensuring a balance between high-participation clients and those underrepresented in the federation. This balance produces more stable and effective training and enables higher performance in challenging settings characterized by both data and resource heterogeneity. Resource heterogeneity often increases the number of underrepresented clients because many client selection strategies prioritize fast and consistently available devices while excluding clients with constrained computational resources from the training process. Although prioritizing such

clients may offer advantages in iid settings, it becomes a challenge under non-IID distributions. In healthcare, these slower clients frequently hold rare classes or essential domain information. Excluding the theme removes critical data from the learning process and degrades the model performance. As a result, when data heterogeneity and resource constraints coexist, traditional client selection methods fail to maintain domain diversity and struggle to achieve robust performance in federated learning.

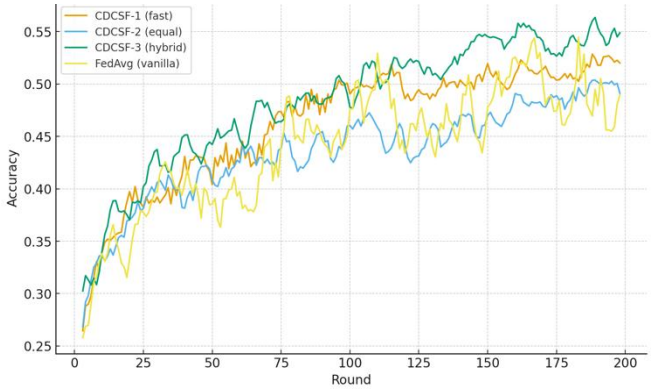

Fig. 7. Comparison of client selection policies on PathMNIST under severe domain shift and non-iid data heterogeneity.

V. Conclusion

In this work, we introduced CDCSF, a domain-aware client selection framework that combines prototype-based clustering, reliability estimation, and fairness participation to guide dynamic selection during training. The framework aims to preserve domain diversity, reduce sampling bias, and ensure that underrepresented clients contribute to training despite their computational capacity. In healthcare settings, data are often limited and heterogeneous. Effective client selection is therefore essential to preserve diversity, improve representativeness, and enhance model robustness. Despite these advantages, CDCSF remains sensitive to clustering hyperparameters, including the number of clusters and update frequency. Inappropriate settings may disrupt the EM optimization process, which may slow model convergence. Furthermore, although prototype-based clustering preserves privacy, it may introduce additional communication overhead. In addition, CDCSF adjusts client weights to mitigate straggler effects. However, extreme system heterogeneity can still delay aggregation and slow training. Overall, CDCSF demonstrates that domain-aware client selection can simultaneously address data heterogeneity and resource constraints in federated healthcare settings. Extending the framework to real-world multi-institutional deployments and validating it under more real-world datasets represent the most critical directions for future research.

V. Conflict of Interest

The author declares that there are no conflicts of interest related to this research.

References

[1] B. Camajori Tedeschini et al., "Decentralized Federated Learning for Healthcare Networks: A Case Study on Tumor Segmentation," IEEE Access, vol. 10, pp. 8693–8708, 2022, doi: 10.1109/ACCESS.2022.3141913.

[2] M. J. Sheller et al., "Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data," Sci. Rep, vol. 10, no. 1, p. 12598, July 2020, doi: 10.1038/s41598-020-69250-1.

[3] W. Zhang et al., "Dynamic Fusion based Federated Learning for COVID-19 Detection," 2020, arXiv. doi: 10.48550/ARXIV.2009.10401.

[4] H. R. Roth et al., "Federated Learning for Breast Density Classification: A Real-World Implementation," in Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning, vol. 12444, S. Albarqouni, S. Bakas, K. Kamnitsas, M. J. Cardoso, B. Landman, W. Li, F. Milletari, N. Rieke, H. Roth, D. Xu, and Z. Xu, Eds., in Lecture Notes in Computer Science, vol. 12444. , Cham: Springer International Publishing, 2020, pp. 181–191. doi: 10.1007/978-3-030-60548-3_18.

[5] L. Li, N. Xie, and S. Yuan, "A Federated Learning Framework for Breast Cancer Histopathological Image Classification," Electronics, vol. 11, no. 22, p. 3767, Nov. 2022, doi: 10.3390/electronics11223767.

[6] J. Wang et al., "A Field Guide to Federated Optimization," 2021, arXiv. doi: 10.48550/ARXIV.2107.06917.

[7] L. Fu, H. Zhang, G. Gao, M. Zhang, and X. Liu, "Client Selection in Federated Learning: Principles, Challenges, and Opportunities," 2022, arXiv. doi: 10.48550/ARXIV.2211.01549.

[8] Y. J. Cho, J. Wang, and G. Joshi, "Client Selection in Federated Learning: Convergence Analysis and Power-of-Choice Selection Strategies," 2020, arXiv. doi: 10.48550/ARXIV.2010.01243.

[9] W. Marfo, D. K. Tosh, and S. V. Moore, "Efficient Client Selection in Federated Learning," 2025, arXiv. doi: 10.48550/ARXIV.2502.00036.

[10] M. Ribero and H. Vikalo, "Communication-Efficient Federated Learning via Optimal Client Sampling," 2020, arXiv. doi: 10.48550/ARXIV.2007.15197.

[11] R. Yan et al., "Label-Efficient Self-Supervised Federated Learning for Tackling Data Heterogeneity in Medical Imaging," IEEE Trans. Med. Imaging, vol. 42, no. 7, pp. 1932–1943, July 2023, doi: 10.1109/TMI.2022.3233574.

[12] Y. Sun, N. Chong, and H. Ochiai, "Feature Distribution Matching for Federated Domain Generalization," 2022, arXiv. doi: 10.48550/ARXIV.2203.11635.

[13] J. Chen, M. Jiang, Q. Dou, and Q. Chen, "Federated Domain Generalization for Image Recognition via Cross-Client Style Transfer," 2022, arXiv. doi: 10.48550/ARXIV.2210.00912.

[14] G. Wu and S. Gong, "Collaborative Optimization and Aggregation for Decentralized Domain Generalization and Adaptation," in 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada: IEEE, Oct. 2021, pp. 6464–6473. doi: 10.1109/ICCV48922.2021.00642.

[15] W. Zhang and X. Li, "Federated Transfer Learning for Intelligent Fault Diagnostics Using Deep Adversarial Networks With Data Privacy," IEEE/ASME Trans. Mechatron., vol. 27, no. 1, pp. 430–439, Feb. 2022, doi: 10.1109/TMECH.2021.3065522.

[16] Y. Tan et al., "FedProto: Federated Prototype Learning across Heterogeneous Clients," 2021, arXiv. doi: 10.48550/ARXIV.2105.00243.

[17] Y. Qiao, H. Q. Le, M. Zhang, A. Adhikary, C. Zhang, and C. S. Hong, "FedCCL: Federated dual-clustered feature contrast under domain heterogeneity," Information Fusion, vol. 113, p. 102645, Jan. 2025, doi: 10.1016/j.inffus.2024.102645.

[18] U. Michieli and M. Ozay, "Prototype Guided Federated Learning of Visual Feature Representations," 2021, arXiv. doi: 10.48550/ARXIV.2105.08982.

[19] X. Mu et al., "FedProc: Prototypical Contrastive Federated Learning on Non-IID data," 2021, arXiv. doi: 10.48550/ARXIV.2109.12273.

[20] F. Sattler, K.-R. Müller, and W. Samek, "Clustered Federated Learning: Model-Agnostic Distributed Multi-Task Optimization under Privacy Constraints," 2019, arXiv. doi: 10.48550/ARXIV.1910.01991.

[21] Y. Zhang, H. Chen, Z. Lin, Z. Chen, and J. Zhao, "FedAC: An Adaptive Clustered Federated Learning Framework for Heterogeneous Data," 2024, arXiv. doi: 10.48550/ARXIV.2403.16460.

[22] G. Long, M. Xie, T. Shen, T. Zhou, X. Wang, and J. Jiang, "Multi-center federated learning: clients clustering for better personalization," World

Wide Web, vol. 26, no. 1, pp. 481–500, Jan. 2023, doi: 10.1007/s11280-022-01046-x.

[23] M. Duan et al., "Flexible Clustered Federated Learning for Client-Level Data Distribution Shift," IEEE Trans. Parallel Distrib. Syst., pp. 1–1, 2021, doi: 10.1109/TPDS.2021.3134263.

[24] T. Nishio and R. Yonetani, "Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge," in ICC 2019 - 2019 IEEE International Conference on Communications (ICC), Shanghai, China: IEEE, May 2019, pp. 1–7. doi: 10.1109/ICC.2019.8761315.

[25] Z. Sasindran, H. Yelchuri, and T. V. Prabhakar, "Ed-Fed: A generic federated learning framework with resource-aware client selection for edge devices," in 2023 International Joint Conference on Neural Networks (IJCNN), Gold Coast, Australia: IEEE, June 2023, pp. 1–8. doi: 10.1109/IJCNN54540.2023.10191316.

[26] M. Cao, Y. Zhang, Z. Ma, and M. Zhao, "C 2 S : Class-aware client selection for effective aggregation in federated learning," High-Confidence Computing, vol. 2, no. 3, p. 100068, Sept. 2022, doi: 10.1016/j.hcc.2022.100068.

[27] F. Lai, X. Zhu, H. V. Madhyastha, and M. Chowdhury, "Oort: Efficient Federated Learning via Guided Participant Selection," in 15th USENIX Symposium on Operating Systems Design and Implementation (OSDI 21), USENIX Association, July 2021, pp. 19–35. [Online]. Available: https://www.usenix.org/conference/osdi21/presentation/lai

[28] C. Li, X. Zeng, M. Zhang, and Z. Cao, "PyramidFL: a fine-grained client selection framework for efficient federated learning," in Proceedings of the 28th Annual International Conference on Mobile Computing And Networking, Sydney NSW Australia: ACM, Oct. 2022, pp. 158–171. doi: 10.1145/3495243.3517017.

[29] T. Huang, W. Lin, W. Wu, L. He, K. Li, and A. Y. Zomaya, "An Efficiency-Boosting Client Selection Scheme for Federated Learning With Fairness Guarantee," IEEE Trans. Parallel Distrib. Syst., vol. 32, no. 7, pp. 1552–1564, July 2021, doi: 10.1109/TPDS.2020.3040887.

[30] Z. Chai et al., "TiFL: A Tier-based Federated Learning System," 2020, arXiv. doi: 10.48550/ARXIV.2001.09249.

[31] W. Huang, M. Ye, Z. Shi, H. Li, and B. Du, "Rethinking Federated Learning with Domain Shift: A Prototype View," in 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada: IEEE, June 2023, pp. 16312–16322. doi: 10.1109/CVPR52729.2023.01565.

[32] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning," 2017, arXiv. doi: 10.48550/ARXIV.1702.07464.

[33] J. Yang *et al.*, "MedMNIST v2 - A large-scale lightweight benchmark for 2D and 3D biomedical image classification," *Sci Data*, vol. 10, no. 1, p. 41, Jan. 2023, doi: 10.1038/s41597-022-01721-8.

[34] Beutel DJ, Topal T, Mathur A, Qiu X, Fernandez-Marques J, Gao Y, et al. Flower: a friendly federated learning research framework. arXiv:2007.14390. 2020. doi:10.48550/ARXIV.2007.14390.