IP-FL: Incentive-driven Personalization in Federated Learning

Ahmad Faraz Khan
Virginia Tech
Blacksburg, USA
ahmadfk@vt.edu

Xinran Wang
University of Minnesota
Minnesota, USA
wang8740@umn.edu

Qi Le
University of Minnesota
Minnesota, USA
le000288@umn.edu

Zain ul Abdeen
Virginia Tech
Blacksburg, USA
zabdeen@vt.edu

Azal Ahmad Khan
University of Minnesota
Minnesota, USA
khan1069@umn.edu

Haider Ali
Virginia Tech
Blacksburg, USA
haiderali@vt.edu

Ming Jin
Virginia Tech
Blacksburg, USA
Jinming@vt.edu

Jie Ding
University of Minnesota
Minnesota, USA
dingj@umn.edu

Ali R. Butt
Virginia Tech
Blacksburg, USA
butta@vt.edu

Ali Anwar
University of Minnesota
Minnesota, USA
aanwar@umn.edu

2025 IEEE International Parallel and Distributed Processing Symposium (IPDPS) | 979-8-3315-3237-6/25/$31.00 ©2025 IEEE | DOI: 10.1109/IPDPS64566.2025.00053

***Abstract*— Federated Learning (FL) is an approach for privacy-preserving Machine Learning (ML), enabling model training across multiple clients without centralized data collection. Existing incentive solutions for traditional Federated Learning (FL) focus on individual contributions to a single global objective, neglecting the nuances of clustered personalization with multiple cluster-level models and the non-monetary incentives such as personalized model appeal for clients. In this paper, we first propose to treat incentivization and personalization as interrelated challenges and solve them with an incentive mechanism that fosters personalized learning. Additionally, current methods depend on an aggregator for client clustering, which is limited by a lack of access to clients' confidential information due to privacy constraints, leading to inaccurate clustering. To overcome this, we propose direct client involvement, allowing clients to indicate their cluster membership preferences based on data distribution and incentive-driven feedback. Our approach enhances the personalized model appeal for self-aware clients with high-quality data leading to their active and consistent participation. Our evaluation demonstrates significant improvements in test accuracy (8–45%), personalized model appeal (3–38%), and participation rates (31–100%) over existing FL models, including those addressing data heterogeneity and personalization.**

***Index Terms*—Federated Learning, Incentivized Learning, Personalized Learning, Machine Learning Systems**

I. Introduction

Training high-quality models using traditional distributed Machine Learning requires data transfer from the data sources to a central location, which raises various communication, computation, and privacy challenges. In response, Federated Learning (FL) [35], [46] has emerged as a solution to train models at the source, reducing privacy issues and addressing the need for high-quality models. However, the success of FL relies on resolving various challenges related to statistical heterogeneity [41], [45], [59], [62], scheduling [5], [20], [32], [37], [39], and incentive distribution [9], [51], [53]. To overcome data heterogeneity challenges, personalized FL (pFL) has emerged as an effective solution to generate separate yet related models [6], [8], [36], [38], [52]. Among pFL techniques, similarity-based approaches that use clustering of clients at the aggregator have gained popularity [11], [44], [49], [55], [58].

However, existing pFL solutions do not include any incentive mechanism, which is crucial in FL to motivate participants to contribute their data and computation resources. Existing incentive mechanisms [9], [21], [25] for traditional FL cannot be applied to pFL techniques because they only consider the performance contribution of clients towards training a single objective. In contrast, clients in pFL contribute towards multiple objectives simultaneously [11], [40], [45], [49], [54], [62]. Furthermore, traditional incentive solutions only provide monetary benefits and do not consider increasing personalized models' appeal as an incentive for encouraging active and reliable participation of clients. Without incentives, participants may provide low-quality data [9], [15], [16], [25], [51] or opt-out from participation[1] [60], leading to poorly performing pFL models [25], [53]. Collaboration fairness [13], [50] can also be ensured by appropriately rewarding contributions and accounting for data heterogeneity [51], [63].

In addition, since existing pFL techniques assume voluntary and consistent participation from clients, the aggregator controls the client selection and training with insufficient knowledge of clients' training capacity, availability, frequency of new incoming data, clustering preferences, and performance requirements from the trained personalized models. These factors can directly influence the motivation of self-conscious clients to participate consistently. In this paper, we show that these factors cause frequent opt-outs from uninterested clients due to uninformed clustering decisions by the server and low personalized model appeal (PMA)[2], which leads to reduced pFL performance. We also show that solving personalization and incentivization as interrelated challenges yields better outcomes for pFL than solving them as separate problems. Incentives can establish a feedback mechanism by providing a cost-benefit analysis guiding clients to make informed decisions in the pFL process. However, this requires new paradigms for clustered

[1]By "opt-out" we mean the clients voluntarily leave FL due to the lack of incentivization.

[2]Akin to global model appeal [7], we propose a new metric to measure the personalized model appeal.

1530-2075/25/$31.00 ©2025 IEEE
DOI 10.1109/IPDPS64566.2025.00053

Authorized licensed use limited to: Florida Atlantic University. Downloaded on September 18,2025 at 15:24:44 UTC from IEEE Xplore. Restrictions apply.

pFL using data distribution information available to clients via their preferences and designing incentive mechanisms for increasing pFL appeal to reduce client opt-outs.

In this paper, we propose IP-FL that combines clustering-based pFL with token-based incentivization. Unlike prior works that control clustering from the server side, IP-FL allows clients to estimate the importance of each cluster and send their preferences for joining them to the aggregator as bids. To identify a cluster's importance for a client we use the importance weight of the cluster model as defined by FedSoft [49]. Clients also use the importance weights for weighted local aggregation during single-shot personalization. This client-driven clustering approach results in accurate clustering because clients can attain a global perspective from their local dataset which is only accessible to them and the importance weights information of each cluster. This allows them to make informed decisions that the server cannot make, resulting in improved PMA and reduced opt-outs. To incentivize clients for consistent participation, IP-FL motivates clients to join clusters with the clients that are most similar to them, maximizing their contribution to the cluster and, in turn, their rewards. Good quality cluster-level models then produce more appealing personalized models for each client. The incentive mechanism treats clients as both providers and consumers. As a consumer, the client tries to attain a certain level of personalized model appeal, so it pays the provider to spend resources to participate in training for the said model in each round. Whereas as a provider, the client earns a profit based on its marginal contribution to training the cluster models. The marginal contribution is calculated with a Shapley Value approximation due to the large computational overhead of the original algorithm [31], [42], [57].

In this work, we propose IP-FL, an incentive-driven pFL method. The key contributions of IP-FL are as follows: First, it integrates client preferences for personalization and provides a feedback mechanism via contribution-based incentives that results in accurate clustering choices resulting in improved personalization. Second, IP-FL introduces novel incentives such as improved personalized models' appeal for clients to prevent opt-outs. Third, IP-FL has the added advantage of creating personalized models for unseen clients with unknown data distributions that perform similarly to seen clients without requiring additional training. Lastly, we show the efficacy of IP-FL with a comprehensive theoretical analysis and rigorous experimental verification, demonstrating its superiority compared to other pFL solutions. In particular, we observe an 8–45% test accuracy improvement of the cluster models, 3–38% improvement in personalized model appeal, and 31–100% increase in the participation rate compared to a wide range of FL and pFL solutions.

II. Background and Related Work

A. Federated Learning Overview

Federated Learning (FL) is an innovative ML paradigm that enables a group of devices or organizations to collaboratively refine a shared model while maintaining the privacy and decentralization of their data [1], [27]. In this approach, a global model θ is initially set up on a central server. Each participant then performs local model updates using their own data, following an optimization approach, typically stochastic gradient descent (SGD) [46]. The local model update is given as $\theta_i \leftarrow \theta_i - \eta \cdot \nabla L(\theta_i, D_i)$, where θ_i indicates local model parameters, η represents the learning rate, and $\nabla L(\theta_i, D_i)$ is the gradient of the loss function. After local training, the participants transmit their updated models to the central server, where they are aggregated using a fusion algorithm such as weighted averaging given as $\theta \leftarrow \frac{1}{N}\sum_{i=1}^{N}\theta_i$. Here, N symbolizes the number of participating devices. This iterative training continues until the global model θ is sufficiently refined for deployment. FL is particularly relevant in domains where privacy and data distribution are key concerns, such as healthcare and finance [2], [3], [17], [24], [28]–[30], [47].

B. Personalized Federated Learning

Cluster-based pFL: Works like FedSoft [49], FedGroup [11], and [54] employ clustering techniques in pFL. FedSoft performs soft clustering by matching client data distributions, whereas FedGroup evaluates client gradient similarities through the Euclidean distance of decomposed cosine similarity. [54] determines personalization-generalization trade-offs via a bi-level optimization problem. Despite these, issues like clustering overhead and client distribution overlap restrictions exist. Other noteworthy models are IFCA [14], focusing on loss-based clustering, and [44], presenting three personalization strategies.

Other pFL models: Meta-learning techniques for rapid personalized model training are found in works like PerFedAvg [12] and regularization models [22], [23]. Multi-task learning models include Ditto [40], FedALA [59], and pFedMe [10]. Additionally, FedFomo [62] introduces adaptive local aggregation for personalization, while FedProx [41] aims to stabilize FL. However, many of these models face challenges in sustaining long-term client engagement and require additional training or restructuring for new clients [34].

C. Incentivized Federated Learning

FAIR [9] and FedFAIM [51] focus on quality and fairness-based incentives, respectively. The reputation-based and reverse auction method is highlighted in [60], and a utility-centric approach in [25]. Yet, these methods don't integrate seamlessly with all pFL models.

Why existing incentive mechanisms cannot be applied directly to pFL frameworks? Standard FL incentives targeting a single global goal [9], [25], [51], [60] may not align with the multifaceted objectives in pFL, such as cluster-based [11], [49], [54] or multi-task models [10], [23], [40]. IP-FL employs clustering for pFL, adjusting cluster memberships every R rounds. It distinguishes itself by establishing cluster boundaries while enhancing shared learning via multiple participation at the client level. Moreover, IP-FL promotes consistent client participation using PMA and an incentive mechanism based on Individual Rationality (IR) from game theory [9], [56], [61].

Authorized licensed use limited to: Florida Atlantic University. Downloaded on September 18,2025 at 15:24:44 UTC from IEEE Xplore. Restrictions apply.

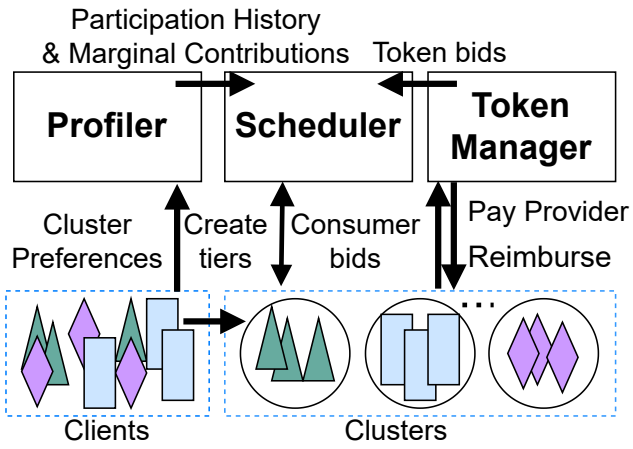


Fig. 1: IP-FL design

III. Problem Formulation

In pFL, each client's natural goal is to develop a model that maximizes its test accuracy. Different clients may have varying thresholds for the minimum accuracy gain required to justify their participation in pFL. We denote this self-defined threshold as ρ_i, where $i \in [N]$. The threshold ρ_i is defined as the test accuracy that client i would achieve if it used the conventional FedAvg approach. Therefore, PMA_i represents the gain in performance from pFL compared to vanilla FL using FedAvg for a client i among N clients. PMA is analogous to GMA from [7], however, our analysis and experiments demonstrate that creating a single global model may not align with the interests of all clients. We formalize the concept of PMA and opt-outs as follows:

$$\mathrm{PMA}_i = f_i(w_k) - \rho_i \mid i \in [N], k \in [K] \tag{1}$$

$$\text{opt-outs} = \sum_{i=1}^{N} H(\mathrm{PMA}_i) \text{ for } i \in [N], H(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ 1 & \text{if } x > 0 \end{cases} \tag{2}$$

Where $f_i(w_k)$ is the test accuracy achieved by pFL for client i with model parameters w_k.

However, pFL faces the challenge of incentive distribution, as this approach necessitates considering the multidimensional nature of personalization. Furthermore, data-shift and feature-shift in client data over time complicate clients' decisions on whether continued participation is beneficial, leading to potential opt-outs. Therefore, we aim to *develop a pFL framework that provides specialized incentives for personalized training, thereby reducing opt-outs and increasing* PMA *to ensure successful collaboration in pFL.*

A. Theoretical foundation

We study the following particular case to develop insights. Suppose there are m clients in total, each observing a set of independent Gaussian observations $z_{i,j} \sim \mathcal{N}(\mu_i, \sigma^2), j = 1, \ldots, n_i$, with a personalized task of estimating its unknown mean $\mu \in \mathbb{R}$. The quality of the learning result, denoted by $\hat{\mu}$, will be assessed by the mean squared error $\mathbb{E}_i(\hat{\mu} - \mu)^2$, where the expectation $\mathbb{E}_i$ is taken with respect to the distribution of client i. It is conceivable that if clients' underlying parameters μ_i's are arbitrarily given, personalized FL may not boost the local learning result. To highlight the potential benefit of cluster-based modeling, we suppose that the m clients can be partitioned into two or more subsets, a common assumption in related works [14], [40], [49]. For simplicity of the problem, we make a case for two subsets: one with m_1 clients, say $T_1 = \{1, \ldots, m_1\}$, and the other with m_2 clients, say $T_2 = \{m_1 + 1, \ldots, m\}$, whose underlying parameters are randomly generated as follows:

$$\mu_i \sim \mathcal{N}(\beta_1, \tau^2) \mid \quad i \in T_1, \quad \mu_i \sim \mathcal{N}(\beta_2, \tau^2) \mid \quad i \in T_2. \tag{3}$$

Here, β_1 and β_2 can be treated as the root cause of two underlying clusters. We will study how the values of sample size n_i, data variation σ, within-cluster similarity as quantified by τ, and cross-cluster similarity as quantified by $|\beta_1 - \beta_2|$ will influence the gain of a client in personalized learning. To simplify the discussion, we will assess the learning quality (based on the mean squared error) of any particular client i in the following three procedures:

Local training: Client i only performs local learning by minimizing the local loss $L_i(\mu) = \sum_{j=1}^{n_i}(\mu - z_{i,j})^2$, and obtains $\hat{\mu}_i = n_i^{-1} \sum_{j=1}^{n_i} z_{i,j}$. Thus, the corresponding error variance is

$$e(\hat{\mu}_i) = \mathbb{E}_i(\hat{\mu}_i - \mu_i)^2 = \frac{\sigma^2}{n_i}. \tag{4}$$

Federated training: Suppose the FL converges to the global minimum of the loss, $\sum_{i=1}^{m} \frac{n_i}{n} L_i(\mu), \quad n \triangleq \sum_{i=1}^{m} n_i$, which can be calculated to be $\hat{\mu}_{\mathrm{FL}} = \sum_{i=1}^{m} \frac{n_i}{n} \hat{\mu}_i$. Consider any particular client i. Without loss of generality, suppose it belongs to cluster 1, namely $i \in T_1$. From the client i's angle, conditional on its local μ_i and assuming a flat prior on β_1 and β_2, client j's μ_j follows $\mu_j \mid \mu_i \sim \mathcal{N}(\mu_i, 2\tau^2)$ for $j \in T_1$ and $j \neq i$, and $\mu_j \mid \mu_i \sim \mathcal{N}(\mu_i + \beta_2 - \beta_1, 2\tau^2)$ for $j \in T_2$. Then, the corresponding error is

$$e(\hat{\mu}_{\mathrm{FL}}) = \mathbb{E}_i(\hat{\mu}_{\mathrm{FL}} - \mu_i)^2 = \left\{ \sum_{j \in T_2} \frac{n_j}{n} (\beta_2 - \beta_1) \right\}^2 + \sum_{j=1,\ldots,m, j \neq i} \left(\frac{n_j}{n} \right)^2 \left(\frac{\sigma^2}{n_j} + 2\tau^2 \right) + \left(\frac{n_i}{n} \right)^2 \frac{\sigma^2}{n_i}. \tag{5}$$

It can be seen that compared with (4), the above FL error can be non-vanishing if $\sum_{j \in T_2} \frac{n_j}{n} (\beta_2 - \beta_1)$ is away from zero, even if sample sizes go to infinity. In other words, in the presence of a significant difference between the two clusters, FL may not bring additional gain compared with local learning.

Cluster-based personalized FL: Suppose our algorithm allows both clusters to be correctly identified upon convergence. Consider any client i. Suppose it belongs to Cluster 1 and will use a weighted average of Cluster-specific models. Specifically, the Cluster 1 model will be the minimum of the loss $\sum_{j \in T_1} \frac{n_j}{n_{\mathrm{T1}}} L_j(\mu), \quad n_{\mathrm{T1}} \triangleq \sum_{j \in T_1} n_j$, which can be

Authorized licensed use limited to: Florida Atlantic University. Downloaded on September 18,2025 at 15:24:44 UTC from IEEE Xplore. Restrictions apply.

calculated to be $\hat{\mu}_{\mathrm{T1}} = \sum_{j \in T_1} \frac{n_j}{n_{\mathrm{T1}}} \hat{\mu}_i$. By a similar argument as in the derivation of (5), we can calculate

$$e(\hat{\mu}_{\mathrm{T1}}) = \sum_{j \in T_1, j \neq i} \left(\frac{n_j}{n_{\mathrm{T1}}}\right)^2 \left(\frac{\sigma^2}{n_j} + 2\tau^2\right) + \left(\frac{n_i}{n_{\mathrm{T1}}}\right)^2 \frac{\sigma^2}{n_i}. \quad (6)$$

The above value can be smaller than that in (4). To see this, let us suppose the sample sizes n_i's are all equal to, say n_0, for simplicity. Then, we have

$$e(\hat{\mu}_{\mathrm{T1}}) = \frac{m_1 - 1}{m_1^2}\left(\frac{\sigma^2}{n_0} + 2\tau^2\right) + \frac{1}{m^2}\frac{\sigma^2}{n_0} = \frac{m_1 - 1}{m_1^2}\left(\frac{\sigma^2}{n_0} + 2\tau^2\right) + \frac{1}{m_1^2}\frac{\sigma^2}{n_0} = \frac{1}{m_1}\frac{\sigma^2}{n_0} + \frac{m_1 - 1}{m_1^2}2\tau^2,$$

which is smaller than (4) if and only if: $\tau^2 < \frac{m_1 \sigma^2}{2n_0}$. We derive the following intuitions from this analysis: **R1.** If the within-cluster bias is relatively small, the number of cluster-specific clients and data noise is large, a client will have personalized gain from collaborating with clients in the same cluster. **R2.** IP-FL's incentive algorithm rewards accuracy improvement reflected in PMA, which directly correlates with reducing within-cluster bias as per the equation above. **R3.** By association, the incentive algorithm motivates clients to join similar clusters which increases cluster homogeneity and reduces the within-cluster bias. We later show the impact of change in performance with an ablation study of IP-FL incentive.

Algorithm 1 IP-FL (Server)

Input: R: Rounds, K: Number of clusters, M_k: Cluster-level model of cluster $k \in K$, N: Number of clients, ζ_a: Available Clients, N_p: Number of clients selected on performance basis, N_r: Number of clients selected randomly, ζ_k: Clients selected for training in cluster $k \in K$, *sort()*:Timsort [48]

1 **for** *each round* $r \in R$ **do**
2 $\zeta_k = SelectClients(r)$ for each cluster $k \in K$
3 **for** *cluster* $k \in K$ **do**
4 Server deploys model M_k for training client in ζ_k
5 Token Manager collects bid payments from all willing clients via Eqn. 9
6 Token manager updates available tokens for round r via Eqn. 10
7 $U_k \leftarrow$ model updates received from clients in ζ_k
8 $M_k = FedAvg(U_k)$ [46]

9 **Function** `SelectClients(r)`
10 **if** $r = 0$ **then**
11 **for** $k = 1$ ***to*** K **do**
12 $\zeta_k^* \leftarrow$ Scheduler selects random clients from ζ_a.
13 **return** ζ_k^*
14 **else if** $r > 1$ **then**
15 **for** $i = 1$ ***to*** N **do**
16 $P_B \leftarrow ClientPreferences(M_1, ..., M_k) \mid \forall k \in [K]$ // from Algo. 2
17 The server calculates marginal contributions ψ_{ki} of each client within its cluster via Shapley Values approximation [26].
18 $S_c = sort(P_B, \psi_{ki})$ //Profiler sorts clients by marginal contributions and preference bids
19 **for** $k = 1$ ***to*** K **do**
20 $\zeta_k^* \leftarrow N_p$ clients selected from S_c and N_r clients randomly from ζ_a by Scheduler.
21 **return** ζ_k^*

IV. Design of IP-FL

In this section, we introduce IP-FL, consisting of three key modules: profiler, token manager, and the scheduler (Figure 1). The profiler computes and maintains client contributions using Shapley Values approximation (lines 24-27 in Algorithm 1), aiding cluster formation. The token manager handles auctions, rewards, and reimbursement transactions (lines 13 & 14). The scheduler selects clients based on bids and contributions, clustering them for improved homogeneity (lines 20 & 27-29). Clients calculate importance weights from cluster models, submit preference bids to the token manager for cluster participation (lines 23-28 in Algorithm 2), and generate personalized models (line 29). Clients seek to maximize profits through IR by choosing clusters where they can contribute the most for maximum rewards.

A. Profiler

At the pFL training's onset, the scheduler forms initial clusters by assigning clients randomly. At each round, clients train the cluster-level model on local data and compute the importance weight for each aggregated cluster model M_k via Equation 7, where v_{ik} is the normalized sum of correctly predicted data points on local dataset D_i. These weights create a personalized model through weighted aggregation via Equation 8. Here, P_{ik} is client i's personalized model and ω_k is cluster k's weight vector. Through client-centric clustering and participation, clients produce personalized models offline to suit dynamic needs unknown to the server. Clients can also decide on training participation based on budget, past rewards, and importance weights. The clients then bid for the desired cluster in the next training round.

$$v_{ik} = n_{ik}/n_k \in [0, 1] \mid k \in [K] \quad (7)$$

$$P_{ik} = \sum_{k=1}^{K} v_{ik} \times (\omega_k) \quad (8)$$

Before the start of the next training round, the profiler calculates client marginal contributions using a Shapley Values approximation [26], providing data quality insights of each client to the scheduler.

B. Token Manager

The token manager coordinates client transactions, operating similarly to a bank executing reward transactions from consumer to provider clients and any reimbursement transactions from providers back to consumer clients. At the beginning of each training round, it conducts an auction for each cluster. Each client i interested in a cluster places its bid using tokens. These bids, represented as τ_p, are deducted from the clients' total tokens, τ_i, as detailed in Equation 9:

$$\tau_i = \tau_i - \tau_p \quad (9)$$

Collected tokens are then added to the Token Manager's overall pool τ_{ar} via Equation 10. Np and N_r represent the

Authorized licensed use limited to: Florida Atlantic University. Downloaded on September 18,2025 at 15:24:44 UTC from IEEE Xplore. Restrictions apply.

number of clients selected based on performance and random selection respectively.

$$\tau_{ar} = \tau_{ar} + (Np + N_r) \times \tau_p \mid r \in [1, R] \tag{10}$$

The token manager's responsibilities also encompass rewards and penalties as reimbursements. It calculates reimbursements using the utility function, which quantifies the average accuracy improvement or decrement of the cluster model M_k over the maximum achieved accuracy in past rounds on the local data of clients in cluster k. Then it reimburses consumer clients and penalizes any degradation in the performance of the provider clients to ensure contribution fairness on a finer granularity for consumer and provider clients. This utility and the subsequent reimbursement are defined in Equations 11 and 12:

$$U_{til} = \frac{\eta \times (\gamma - \min(\gamma, \max(0.0, \frac{(Acc_{kr} - Acc_{kmax})}{Acc_{kmax}})))}{\gamma} \mid \eta \in [0,1], \gamma \in [0,1] \tag{11}$$

$$\tau_i = \tau_i - \tau_{ar} \times U_{til} \mid U_{til} \in [0, \gamma], \forall i \in [N], \forall r \in [1, R] \tag{12}$$

Acc_{kr} is the cluster-level model accuracy in round r, Acc_{kmax} is the maximum cluster-level accuracy achieved until this round, η and γ represent token limitations and accuracy improvement thresholds and τ_{ar} is the tokens collected from clients in round r. Our approach is inspired by [21], but it's adapted to account for the impractical assumption of having an IID dataset at the aggregator that perfectly corresponds to the client data distribution within the cluster. Instead, we rely on the local client dataset. Post-reimbursement, the token manager distributes rewards. Provider clients are ranked based on their contributions and participation, and rewards are allocated via Equation 13:

$$\tau_i = \tau_i + \text{sort}(\psi_{ki}, \Omega_{ki}) \times \frac{\tau_{ar}}{N_r \times \frac{(N_r+1)}{2}} \mid \forall k \in [K], \forall i \in [N], \forall r \in [R] \tag{13}$$

In Equation 13, ψ_{ki} represents client contributions, and Ω_{ki} tracks client participation. β normalizes token distribution based on provider ranks α in Equation 13. τ_i are client-owned tokens, and τ_{ar} are tokens available for distribution by the Token Manager. Through reimbursements to consumers and payments to providers, IP-FL incentivizes personalized learning, improving PMA, reducing opt-outs, and ensuring fair client incentives based on contributions to enhance personalized learning outcomes.

C. Scheduler

The scheduler employs the $SelectClients(r)$ function detailed in Algorithm 1 to choose clients for each round r. It collects preference bids P_B from the token manager and marginal contributions ψ_{ki} from the profiler for every client $i \in N$ in cluster $k \in K$. Here, N and K denote the total counts of clients and clusters, respectively. With this data, the scheduler groups clients by similar preference bids, arranges them by their marginal contributions, and subsequently picks N_p from this ordered list and N_r at random. These quantities, N_p and N_r, are adjustable parameters. Following the methodology in [37], we prioritize exploration in early rounds, initializing N_r to 20% of all clients and decrementing it based on the count of remaining unexplored clients. The reduction strategy, which caps the minimum N_r at 5%, guarantees an effective exploration technique. Random selection of N_r clients mitigates accuracy bias, echoing techniques from prior studies [4], [21], [33], [46]. By clustering clients with similar preferences, the scheduler reduces within-cluster bias and enhances within-cluster uniformity, culminating in a cluster model that truly mirrors its constituent clients. Section III delves into the significance of this in boosting the PMA.

Algorithm 2 IP-FL (Client)

Input: T_h: Importance weight threshold, K: Number of clusters, M_k: Cluster-level model of cluster k, D: Local dataset of client

Function `ClientPreferences`$(M_1, ..., M_k)$
  for *each cluster* $k \in K$ **do**
    for *each data point* $d \in D$ **do**
      The client i computes v_{ik} importance weight T_h of M_k model for each data point d via Eqn. 7
    if $v_k > T_h$ **then**
      Client includes cluster k in their preference list P_B
  The client generates personalized model P_{ik} via Eqn. 8
  return P_B

V. Convergence Analysis of IP-FL

This section explores the convergence properties of IP-FL, showcasing its robustness and efficiency. Beginning with foundational propositions, we establish the basis for cluster formation in IP-FL. Subsequent theorems delve into the convergence of personalized and global models across various data distributions, highlighting IP-FL adaptability for achieving stable and optimal parameters. Proofs for all propositions and theorems are detailed below.

Theorem 1 (FL Convergence [19], [27])**.** *For a set of clients $\{c_i\}_{i=1}^N$ participating in the IP-FL framework with convex and β-smooth local loss functions $\{L_i\}_{i=1}^N$, the global model M is updated by a weighted aggregation of local models using a personalized, diminishing learning rate η_t. If $\sum_{t=1}^{\infty} \eta_t = \infty$ and $\sum_{t=1}^{\infty} \eta_t^2 < \infty$, the sequence of global models $\{M_t\}$ generated by IP-FL converges to a global minimizer M^* of the weighted average loss function $L(M) = \sum_{i=1}^N w_i L_i(M)$.*

Theorem 2. *Let $\{\mathcal{C}_k\}_{k=1}^K$ be clusters formed by the IP-FL algorithm, where clients in each cluster $\mathcal{C}_k$ have statistically similar data distributions, up to a threshold δ. If the data in each cluster follows a Gaussian distribution and the loss function $L(\theta)$ is convex, then the IP-FL algorithm converges to a stable set of parameters θ^* for the global model, with each cluster converging to stable parameters θ_k^*.*

Proof. We divide the proof into three parts, as given below.

1. Cluster Convergence: Given the objective function for IP-FL with m clients observing Gaussian data $z_{i,j} \sim N(\mu_i, \sigma^2)$, the global loss is $L(\mu) = \sum_{i=1}^m \frac{n_i}{n} \sum_{j=1}^{n_i} (\mu - z_{i,j})^2$, where μ is the federated global mean estimate, and μ_i is the local mean for

Authorized licensed use limited to: Florida Atlantic University. Downloaded on September 18,2025 at 15:24:44 UTC from IEEE Xplore. Restrictions apply.

client i. Clients are assigned to a cluster k minimizing local loss (i.e $k = \arg\min_k \sum_{j=1}^{n_i} (\beta_k^{(t)} - z_{i,j})^2$, where $\beta_k^{(t)}$ is the cluster centroid. The centroid update rule is $\beta_k^{(t+1)} = \frac{\sum_{i \in C_k^{(t)}} n_i \mu_i}{\sum_{i \in C_k^{(t)}} n_i}$. Since clients and clusters are finite, the monotonic decrease of L ensures convergence when $C_k^{(t)} = C_k^{(t+1)}$ for all k beyond a certain t.

2. Statistical Similarity and Bias Reduction: To establish within-cluster similarity, let $\mathcal{P}$ be a statistical similarity index, such that:

$$|\mathcal{P}(D_i) - \mathcal{P}(D_j)| < \delta \quad \forall i, j \in C_k.$$

The clustering minimizes within-cluster sum of squares (WCSS):

$$\text{minimize}_k \sum_{k=1}^{K} \sum_{i \in C_k} \|x_i - \mu_k\|^2.$$

In IP-FL, $\|x_i - \mu_k\|^2$ corresponds to similarity weight υ_{ik}, with $\upsilon_{ik} \to 1$ implying $\|x_i - \mu_k\|^2 \to 0$. For clients i, j in the same cluster, $\upsilon_{ik}, \upsilon_{jk} \to 1$ implies statistical similarity to μ_k and thus to each other, reducing within-cluster bias: $\text{Bias}(C_k) < \text{Bias}(System)$.

3. Convergence of Parameters: The convex loss $L(\theta)$ satisfies:

$$L(\lambda\theta + (1-\lambda)\theta') \leq \lambda L(\theta) + (1-\lambda)L(\theta'), \quad \forall \lambda \in [0,1].$$

The parameter update is $\theta^{(t+1)} = \theta^{(t)} - \eta_t \nabla L(\theta^{(t)})$, where η_t is the learning rate. By the descent lemma for β-smooth $L(\theta)$,

$$L(\theta^{(t+1)}) \leq L(\theta^{(t)}) - \eta_t \left(1 - \frac{\beta\eta_t}{2}\right) \|\nabla L(\theta^{(t)})\|^2,$$

with $0 < \eta_t < \frac{2}{\beta}$, this guarantees a decrease in L. Assuming $\sum_{t=1}^{\infty} \eta_t = \infty$ and $\sum_{t=1}^{\infty} \eta_t^2 < \infty$, $\{\theta^{(t)}\}$ approaches θ^* where $\nabla L(\theta^*) = 0$. As $t \to \infty$, $\|\theta^{(t+1)} - \theta^{(t)}\| \to 0$ and $\nabla L(\theta^{(t)}) \to 0$, indicating convergence to a stable parameter set θ^* for the global model. □

Theorem 3. *Under the assumption of Theorem 2, there exists a finite number of training iterations T such that for all $t \geq T$, and for every $\epsilon > 0$, the performance metric PMA_i for each client i is within ϵ of its optimal value PMA_i^*.*

Proof. Let $C = \{C_1, C_2, \ldots, C_k\}$ represent the clusters. Each client i updates its model parameters θ_i based on local data D_i and the aggregated model of its cluster. The update at each iteration t is given by:

$$\theta_i^{(t+1)} = \theta_i^{(t)} - \eta_i \nabla L_i(\theta_i^{(t)}, D_i)$$

where η_i is the learning rate, and ∇L_i is the gradient with respect to θ_i. As θ_i approaches its minimum, $\nabla L_i(\theta_i^{(t)}, D_i) \to 0$, reducing $|\theta_i^{(t+1)} - \theta_i^{(t)}|$ for large t. Thus, $\{\theta_i^{(t)}\}$ is Cauchy and converges to θ_i^*. The aggregation combines individual client models into a cluster model, which each client personalizes as:

$$\theta_i^{(t+1)} = \sum_{k=1}^{K} w_{ik} \theta_k^{(t)},$$

where w_{ik} is the weight of the cluster model $\theta_k^{(t)}$ for client i. With proper adjustment of w_{ik} and iterative updates, $\theta_i^{(t)}$ converges to θ_i^* maximizing PMA_i.

Thus, for given $\epsilon > 0$, there exists a finite T such that:

$$|\text{PMA}_i^{(t)} - \text{PMA}_i^*| < \epsilon \quad \forall t \geq T.$$

The inequality follows from the convergence of $\theta_i^{(t)}$ and the continuity of PMA_i with respect to θ_i. □

VI. Evaluation

A. Experimental Setup

We use NVIDIA RTX 3070 GPUs to evaluate IP-FL and other pFL methods across four datasets. Our CNN model (32x64x64 convolutional with 3136x128 linear layer parameters) was designed for efficient training on client devices with limited resources, aligning with Cross-Device FL settings [27].

1) Datasets: **CIFAR10 Data:** We utilized the CIFAR10 dataset from FedSoft [49], consisting of 32x32 RGB images across 10 categories. We emulated data heterogeneity through 10:90, 30:70, linear, and random partitions, dividing data into two clusters, D_A and D_B. In the 10:90 partition, 50 clients received 90% of their training data from D_A and 10% from D_B, and vice versa for the remaining 50 clients. The 30:70 partition followed a similar distribution pattern but with respective 30% and 70% data splits.

EMNIST Data: The EMNIST dataset features 28x28 grayscale images representing 52 classes (26 lowercase and 26 uppercase letters). We applied 10:90 and 30:70 partitions along with linear and random data distributions. In the linear partition, each client k received $(0.5 + k)\%$ of their data from D_A and $(99.5 - k)\%$ from D_B. In the random partition, clients were assigned data based on a randomly generated mixture vector from a Uniform(0, 1) distribution. The EMNIST dataset was divided into K distinct clusters to ensure data exclusivity across clusters.

CIFAR100 Data: We employed the CIFAR100 dataset, which comprises 32x32 RGB images spanning 100 classes. To simulate data heterogeneity, we adopted the same partition strategies (10:90, 30:70, linear, and random) as used for CIFAR10. In the 10:90 partition, half of the clients predominantly received 90% of their training data from cluster D_A and 10% from D_B, while the other half received the inverse allocation. The 30:70 partition was similarly configured with the corresponding percentage splits.

Tiny ImageNet Data: The Tiny ImageNet dataset consists of 64x64 RGB images categorized into 200 classes. We partitioned the data into 10:90, 30:70, linear, and random splits. In the linear partition, each client k was allocated a linearly varying proportion of data from clusters D_A and D_B, while in the random partition, data was assigned based on a mixture vector drawn from a Uniform(0, 1) distribution. This distribtuion ensured that the heterogeneity in Tiny ImageNet mirrored that of the other datasets.

CIFAR10 (Data Drift): This adaptation of CIFAR10 simulates data drift by altering the distribution between training

Authorized licensed use limited to: Florida Atlantic University. Downloaded on September 18,2025 at 15:24:44 UTC from IEEE Xplore. Restrictions apply.

TABLE I: Test accuracy with CIFAR10 dataset

	10:90				30:70			
	IP-FL		FedSoft		IP-FL		FedSoft	
	c0	c1	c0	c1	c0	c1	c0	c1
θ_0	**63.7**	41.3	48.9	49.5	58	57.7	48	48.4
θ_1	43.7	**63.8**	**50.7**	**49.6**	**58.6**	**58.8**	**50**	**50**

TABLE II: Clusters models accuracy with CIFAR10 (data drift)

	10:90		30:70		Linear		Random	
	c0	c1	c0	c1	c0	c1	c0	c1
IP-FL	**62.7(0)**	**71(1)**	**53.3(0)**	**61.9(1)**	**61.7(0)**	**59.4(1)**	**66.9(0)**	**58.4(1)**
FedSoft	32.5(0)	38.6(1)	20.3(0)	23.6(0)	34.42(1)	49.6(1)	21.6(1)	33.12(1)
IFCA	30.5(0)	36.2(0)	18.8(0)	20.4(0)	31.2(1)	46.3(1)	18.7(1)	31.24(1)
FedEM	29.7(0)	37.6(0)	17.3(0)	20.6(0)	30.6(1)	45.8(1)	18.4(1)	30.4(1)

and testing data across clients. In the 10:90 partition, for example, 50 clients had 90% of their training data and 10% of their testing data from D_A, and the reverse from D_B, mimicking dynamic changes in client data over time.

EMNIST (Data Drift): Similar to the CIFAR10 data drift scenario, this adaptation of EMNIST modifies training and testing distributions to reflect data drift. However, for this dataset D_A comprises 26 lowercase letters, and D_B includes 26 uppercase letters, differentiating training and testing datasets to emulate evolving data environments.

EMNIST (Feature Shift): In this adaptation of EMNIST, D_A includes standard EMNIST images, while D_B contains images rotated by 90 degrees, introducing a feature shift that impacts the input data structure.

We use the CIFAR10 (Data Drift) and EMNIST (Data Drift) datasets to present a scenario of data drift at clients or induction of new clients in pFL whose data distributions are unknown. Data drift is a common problem in which there is a mismatch between training and test data, which can lead to significant performance degradation and system inefficiencies [18], [43]. This presents new challenges of accurate clustering and the generation of personalized models for less familiar clients. We observe through empirical evaluation with Synthetic datasets that this can lead to 100% opt-outs from these clients if we use conventional server-driven pFL methods.

2) Focus of Evaluation: First, we compare the clustering ability of IP-FL with a recent clustering-based pFL algorithm [49]. Second, we compare IP-FL with other non-clustering pFL models with a simple test accuracy evaluation. Taking it one step further, we provide a comparison of IP-FL and other clustering and non-clustering pFL models in terms of opt-out reduction and PMA maintenance. Lastly, we show that including client preferences while clustering yields better personalization results because clients can make decisions based on knowledge restricted to the aggregator server. We evaluate all algorithms using the datasets that were employed in their original publications. Thus ensuring that we highlight the strengths of each algorithm, rather than skewing the evaluation towards our methodology.

B. Test Accuracy Performance Comparison

1) Clustering-based pFL models: We evaluate cluster-level model performance on holdout datasets from cluster distributions (D_A and D_B), comparing IP-FL with recent cluster-based pFL algorithm FedSoft on CIFAR10, matching experiment settings from [49] with 100 clients, batch size 128, learning rate $\eta = 0.01$, and 300 training rounds. Table I presents test accuracy for two data partitions: **10:90** and **30:70**, using IP-FL. Notably, IP-FL outperforms FedSoft in the 10:90 partition, where each cluster represents a singular distribution. Clients with more θ_0 data train in cluster c_0 with 63.68% accuracy, and clients with more θ_1 data prefer cluster c_1 with 63.82% accuracy. In contrast, FedSoft scores lower accuracies of 50.7% and 49.6% in the 10:90 partition, facing challenges with data partition variability and representing singular distributions. FedSoft's models c_0 and c_1 show similar performance across θ_0 and θ_1 data, indicating its personalization is more effective when the majority of clients have share data, hence struggling with individual distribution representation and underperforming in non-IID scenarios. The 30:70 partition shows diminished performance, attributed to less data heterogeneity, which constrains personalization-driven accuracy gains. In Tables II and III, we compare IP-FL with three clustering-based pFL algorithms: FedSoft, IFCA, and FedEM [45]. IP-FL consistently outperforms all three algorithms, supporting its effectiveness.

2) Non-clustering pFL models: Table IV shows the test accuracy comparison of IP-FL with other recent pFL algorithms. Some pFL models can perform well for individual partitions such as Ditto for 10:90, FedProx and FedALA for Linear, and PerFedAvg for Random, however, IP-FL consistently outperforms for all data partitions.

C. Evaluation of Opt-outs and PMA

Figure 2 shows the CDF of PMA for clients using different datasets and pFL methods. IP-FL excels, especially for the 10:90 partition where data heterogeneity is most prominent (Figure 2a). In contrast, the EMNIST dataset which features a broader class distribution per client allows FedAvg to perform relatively well, suggesting limited scope for further improvement via personalization. IP-FL enhances PMA, particularly in the 10:90 and 30:70 partitions where other pFL solutions struggle. Table V shows a highly heterogeneous scenario with 52 clusters and 4 classes per client, Ditto and FedProto perform well, however IP-FL outperforms by 15% in PMA. Opt-out ratios for FedProx, FedALA, and PerFedAvg are 0.64, 0.31, and 0.68, respectively, with no opt-outs for Ditto, FedFomo, and IP-FL, showcasing IP-FL's effectiveness in reducing opt-outs and improving PMA under high data heterogeneities.

D. IP-FL performance for multiple clusters

1) On EMNIST dataset: IP-FL outperforms other FL personalization algorithms in heterogeneous cases when the clients and dataset are divided between 2 distributions (DA & DB). To analyze the reliability of IP-FL it is also evaluated with

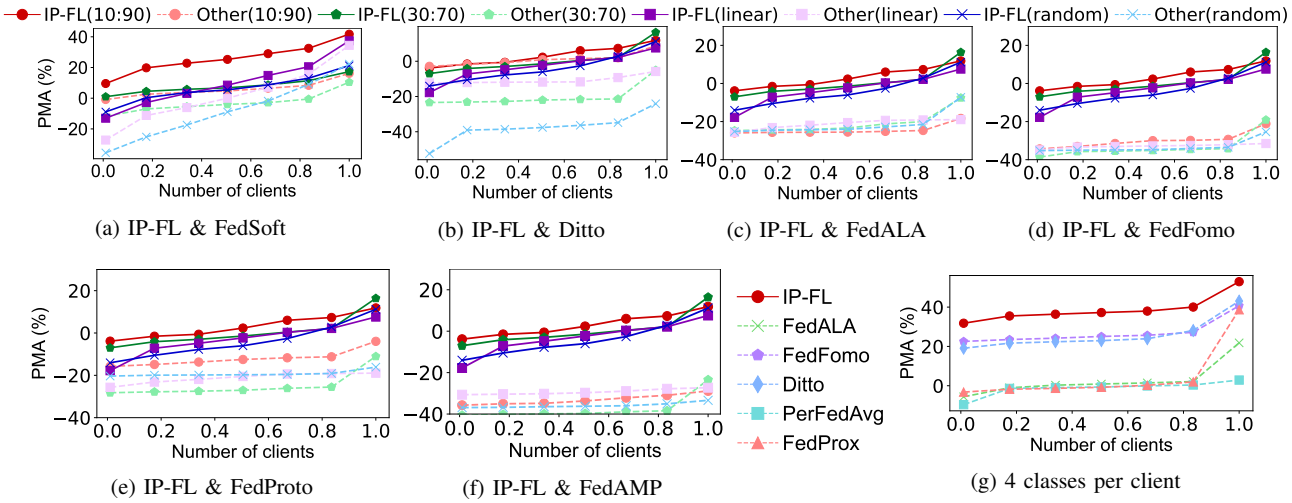
Authorized licensed use limited to: Florida Atlantic University. Downloaded on September 18,2025 at 15:24:44 UTC from IEEE Xplore. Restrictions apply.

TABLE III: Clusters models accuracy with CIFAR10 dataset

	10:90				30:70			
	θ_0		θ_1		θ_0		θ_1	
	c0	c1	c0	c1	c0	c1	c0	c1
IP-FL	**63.7**	41.3	43.7	**63.8**	58	57.7	**58.6**	**58.8**
FedSoft	48.9	49.5	**50.7**	**49.6**	48	48.4	**50**	**50**
IFCA	46.4	44.3	**47**	**45**	46	44	**47.4**	**46.5**
FedEM	45.7	46.5	**46.8**	**47.1**	43.8	45.2	**46.3**	**47.4**

TABLE IV: Test accuracy of pFL methods with EMNIST dataset

Partitions	Ditto	FedProx	FedALA	PerfFedAvg	FedProto	IP-FL
10:90	85.8±4.8	75.2±4.8	75.5±4.7	87.5±3.8	72±1.4	**87.5±3.7**
30:70	76±4.5	79.7±4	78.4±3.2	76.6±3.9	59.7±4.7	**85.1±3.4**
Linear	75.3±5.1	82.8±2.7	82±3.6	80.8±3.5	62.6±4.9	**83.4±4.9**
Random	77.8±6.8	80.9±4.4	79±5.1	83.3±5.2	68.4±5.7	**86.2±4.3**



(a) IP-FL & FedSoft (b) IP-FL & Ditto (c) IP-FL & FedALA (d) IP-FL & FedFomo

(e) IP-FL & FedProto (f) IP-FL & FedAMP (g) 4 classes per client

Fig. 2: CDF of clients' PMA for different datasets and methods

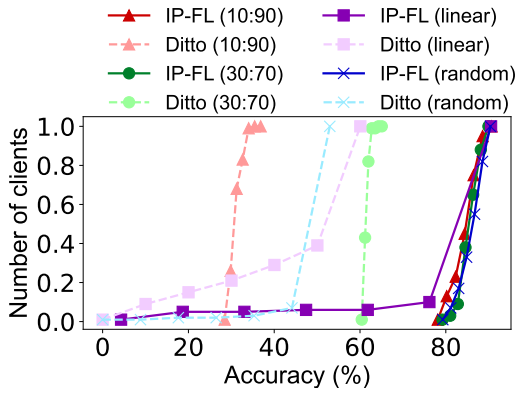


Fig. 3: CDF of test accuracy for IP-FL and Ditto on 4+ distributions using EMNIST Dataset

multiple cluster distributions within a dataset. In this particular experiment, we form 4 different distributions (DA, DB, DC, and DD) on the EMNIST dataset. EMNIST dataset has a total of 56 classes, thus each of the 4 distributions gets 25% of total available classes. DA has the first 13 classes, DB has the next 13, and so on. Figure 3 shows the CDF of test accuracy for all personalized models at clients. IP-FL outperforms Ditto for all partition types. For the linear partition less than 10% clients have lower than 80% accuracy and we attribute this as an outlier due to the partition type where dividing the data linearly some clients get very few data samples. This trend can also be seen with Ditto for the linear partition.

TABLE V: Multiple clusters performance: EMNIST (4 classes per client)

pFL Method	FedAvg	Ditto	FedProx	FedALA	PerFedAvg	FedFomo	IP-FL
Personalized Accuracy	82.8±4.2	90.8±2.1	61.2±2.9	62.2±3.9	60±1.9	86.6±1.6	**98.6±1.2**
Optouts	-	0	0.64	0.31	0.68	0	**0**
Average PMA	-	24.5±4.2	0.6±5.1	1.5±4.74	-0.6±1.5	25.9±3.6	**37.9±3.4**

We further increase the number of clusters and the heterogeneity of the dataset to check IP-FL's robustness for multiple clusters. Table V shows the complete opt-out, test accuracy, and PMA results for the test conducted with the EMNIST dataset with 4 classes per client forming at least **13 clusters**. Some pFL algorithms such as FedFomo and Ditto have similar performance compared to IP-FL in terms of opt-outs. However, IP-FL outperforms all of them in terms of PMA and test accuracy.

2) On CIFAR10 dataset: We also test highly heterogeneous data conditions where each client has only 2 classes per client forming at least **5 clusters**. We use the learning rate = 0.01, batch size = 128, and global epochs = 150 with the CIFAR10 dataset and the same CNN model mentioned in the main paper. Figure 4 shows the PMA for different pFL algorithms. IP-FL outperforms all other pFL algorithms by approximately 50% and FedFomo by 30%. This goes to show that IP-FL performs even better in highly heterogeneous conditions where

Authorized licensed use limited to: Florida Atlantic University. Downloaded on September 18,2025 at 15:24:44 UTC from IEEE Xplore. Restrictions apply.

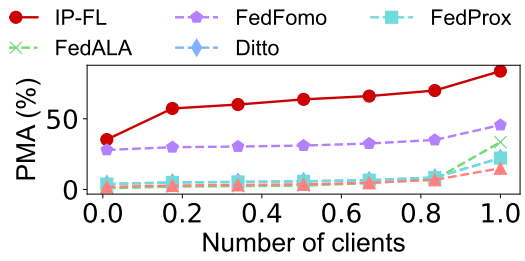


Fig. 4: CDF of clients' PMA for IP-FL and pFL algorithms on CIFAR10 Dataset with 2 classes per client

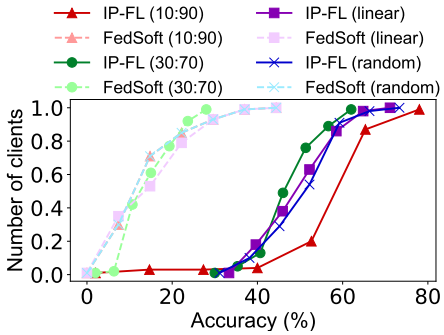


Fig. 5: IP-FL and FedSoft with CIFAR10 (data drift) data

personalization is used to tackle data heterogeneity.

E. IP-FL performance with data drift

We use the CIFAR10 (Data Drift) and EMNIST (Data Drift) datasets to present the data drift at the clients or induction of new clients in pFL whose data distributions are unknown. Data drift causes a mismatch between training and test data and presents new challenges of accurate clustering and the generation of personalized models for less familiar clients. We observe through empirical evaluation with Synthetic datasets that this can lead to 100% opt-outs from these clients if we use conventional server-driven pFL methods. However, we observe that by the induction of client preferences guided by contribution-based incentives in clustering, IP-FL can do accurate clustering and generate appealing models for new clients and changing data distribution at clients.

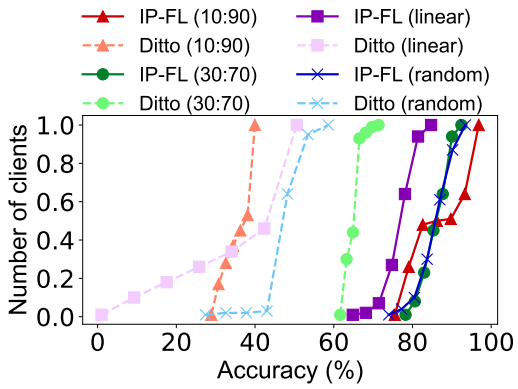


Fig. 6: CDF of clients' personalized model test accuracy for IP-FL and Ditto on EMNIST (data drift)

1) On CIFAR10 (data drift) dataset: Cluster-based pFL methods cluster all heterogeneous clients within different clusters so each cluster has homogeneous clients with similar data distribution in it. So we compare the client-preference driven clustering design of IP-FL with FedSoft using the CIFAR10 (data drift) Data. For this experiment, we use the same configurations described in Section VI-B for the CIFAR10 dataset evaluation. We perform training for 500 rounds with both IP-FL and FedSoft.

Results in Table VI show the cluster-level model test accuracies for IP-FL and FedSoft with CIFAR10 (data drift) data. IP-FL accurately differentiates between clients of different distributions. This is visible by the accuracy difference of each cluster-level model on different distributions. For example on the 10:90 partition $c1$ model has a 70.96% accuracy on the θ_1 distribution and has 2.56% accuracy on θ_0 which indicates that cluster-level model $c1$ trains with clients that have the majority of their training data from θ_1. Similarly, $c0$ trains with clients that have their majority of training data from θ_0 and has an accuracy of 62.68%. For the linear partition, IP-FL $c0$ cluster-level model has an accuracy of 61.66% for distribution θ_0 and 59.44% accuracy for distribution θ_1. However, for the linear partition with FedSoft, both $c0$ and $c1$ perform best on only one distribution θ_1 with accuracy 34.42% and 49.62%. IP-FL outperforms FedSoft in terms of accuracy for each partition and can distinguish between different distributions accurately.

IP-FL's ability to perform accurate clustering also leads to improvements in personalized model test accuracy, even with drift in client data or the inadvertent addition of a new client to the wrong cluster. In Figure 5, we observe the CDF of clients' test accuracy after 500 training rounds. IP-FL's robustness to varying client data is evident, while FedSoft, relying on the server's perspective without access to client data, struggles to make precise clustering decisions.

2) On EMNIST (data drift) dataset: We also compare the performance of IP-FL with non-clustering state-of-the-art pFL methods such as Ditto [40] for the data drift conditions. Since Ditto leverages multi-task learning and does not perform clustering, we only compare the test accuracy of personalized models on the EMNIST dataset for IP-FL and Ditto in Figure 6. IP-FL outperforms Ditto significantly, especially for highly heterogeneous data partitions such as 10:90 and linear. The reason for this performance improvement from IP-FL is that, unlike Ditto, IP-FL provides autonomy to clients to personalize according to their own goals. With Ditto, the goal of each client, which consists of their private data, is hidden from the aggregator server due to privacy reasons, and this affects the quality of the personalized models.

Authorized licensed use limited to: Florida Atlantic University. Downloaded on September 18,2025 at 15:24:44 UTC from IEEE Xplore. Restrictions apply.

TABLE VI: Test accuracy with CIFAR10 (Data Drift)

	IPFL								FedSoft							
	10:90		30:70		linear		random		10:90		30:70		linear		random	
	c0	c1	c0	c1	c0	c1	c0	c1	c0	c1	c0	c1	c0	c1	c0	c1
θ_0	**62.78**	2.56	**53.28**	34.32	**61.66**	12.08	**66.9**	19.48	**32.5**	13.6	**20.28**	**23.58**	8.48	2.82	16.18	0.28
θ_1	1.5	**70.96**	30.38	**61.94**	30.94	**59.44**	19.12	**58.42**	11.76	**38.62**	0.18	0.08	**34.42**	**49.62**	**21.62**	**33.12**

TABLE VII: PMA and opt-outs with EMNIST (Feature Shift)

pFL algo	FedAvg	Ditto	FedProx	FedALA	FedFomo	FedProto	PerfFedAvg	IP-FL
Pers. Acc.	75.7±3	83.7±2	78.8±14	83.3±3	63.9±10	46.5±21	83.7±3	**84.3±2**
Optouts	-	0	23	0	100	0	0	**0**
Avg. PMA	-	7.6	3	7.2	-17	-48.6	7.7	**8.6**

F. IP-FL performance on complex datasets

We expanded our experimental evaluation to include more complex datasets like CIFAR-100 (100 classes) and Tiny ImageNet (200 classes). These datasets introduce higher complexity and diversity, requiring models to generalize across a broader range of classes and distributions. Tables VIII and IX show IP-FL's performance on these datasets compared to state-of-the-art pFL methods under various data partition strategies using the same experimental settings defined in Section VI-A.

TABLE VIII: Performance results on CIFAR-100.

pFL algo	10:90	30:70	Linear	Random
Ditto	57.5±3.2	51.0±3.0	50.5±3.4	52.1±4.6
FedProx	50.5±3.2	53.4±2.7	55.5±1.8	54.0±2.9
FedALA	50.7±3.1	52.5±2.1	54.9±2.4	52.9±3.4
PerfFedAvg	58.7±2.5	51.3±2.6	54.1±2.3	55.8±3.4
FedProto	48.2±0.9	40.0±3.1	42.0±2.7	45.8±3.8
IP-FL	**58.7±2.5**	**57.0±2.3**	**55.8±3.3**	**57.8±2.9**

TABLE IX: Performance results on Tiny ImageNet.

pFL algo	10:90	30:70	Linear	Random
Ditto	35.2±2.0	31.2±1.8	30.8±2.1	31.9±2.8
FedProx	30.8±2.0	32.8±1.6	33.9±1.1	33.1±1.9
FedALA	30.9±1.9	32.1±1.3	33.6±1.5	32.3±2.1
PerfFedAvg	35.9±1.6	31.0±1.6	33.2±1.4	34.0±2.1
FedProto	29.5±0.6	24.3±2.0	25.7±1.9	27.8±2.3
IP-FL	**35.9±1.5**	**35.0±1.4**	**34.1±2.0**	**35.4±1.8**

In CIFAR-100 under the 10:90 partition, IP-FL achieves an accuracy of 58.7%, which matches the best-performing method, PerfFedAvg, and significantly surpasses others like FedProx and FedProto. For Tiny ImageNet under the 30:70 partition, IP-FL achieves 35%, outperforming Ditto and other baselines by a margin of at least 2%. Similar results can be observed for other partitions as well, where IP-FL outperforms other pFL methods even with complex datasets.

In addition, we would like to highlight that among these, IP-FL is the first and only method that does not assume voluntary or forced client participation; instead, it conjoins pFL with incentives to provide a viable solution for realistic scenarios.

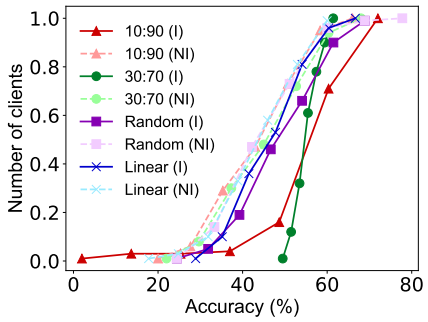


Fig. 7: IP-FL with and without incentive (I/NI)

G. IP-FL performance with data feature-shift

We compared IP-FL to other pFL algorithms in non-IID data scenarios using a feature shift technique, with identical hyperparameters and the EMNIST (feature-shift) dataset. Results in Table VII show that IP-FL that IP-FL surpasses other pFL models, improving test accuracy by 1–38% and PMA by 12-118% with feature-shifted non-IID data.

TABLE X: PMA and opt-outs with EMNIST (40:60) dataset

pFL Method	FedAvg	Ditto	FedProx	FedALA	PerFedAvg	FedFomo	IP-FL
Pers. Acc.	76.2 ± 0.5	77.9 ± 1.1	82.2 ± 0.5	74.6 ± 3.5	79.1 ± 2.4	47.7 ± 14	**85.1 ± 2.1**
Optouts	-	0	26	0	100	0	**0**
Average PMA	-	1.7	5.9	0.14	2.9	-28.6	**8.86**

H. Opt-out results for slightly non-IID EMNIST dataset

We also evaluate the 40:60 partition using the EMNIST dataset, following the parameters outlined in the section VI-C. The results for this are shown in Table X. Notably, in this new partition setting, our proposed IP-FL algorithm performs better compared to other algorithms under consideration. The 40:60 partition is slightly heterogeneous compared to the complete IID case of 50:50 which is why we see an improved performance with personalization particularly with IP-FL.

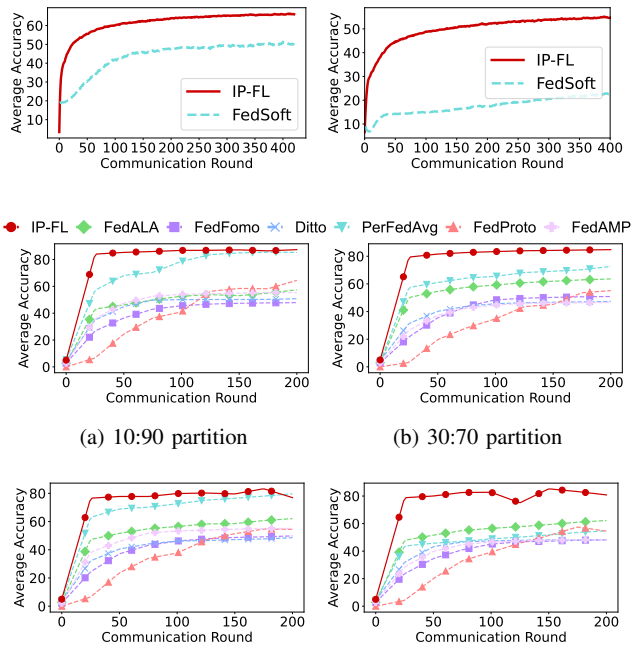
TABLE XI: Test accuracy of ablation study with Incentive (I) and without incentive (NI)

	10:90		30:70		linear		random	
	c0	c1	c0	c1	c0	c1	c0	c1
IP-FL (I)	58.62(0)	67.4(1)	51.12(0)	64.06(1)	66.2(0)	57.02(1)	64.54(0)	56.86(1)
IP-FL (NI)	49.92(1)	52.8(1)	48.9(0)	44.44(1)	50.82(1)	45.92(1)	57.42(1)	46.76(1)

I. Ablation study: pFL with Incentive in IP-FL

We perform an ablation study with the incentive component of IP-FL on the CIFAR10 (data drift) dataset for 200 rounds

Authorized licensed use limited to: Florida Atlantic University. Downloaded on September 18,2025 at 15:24:44 UTC from IEEE Xplore. Restrictions apply.



(a) 10:90 partition (b) 30:70 partition

(c) Linear partition (d) Random partition

Fig. 8: Convergence Speed: IP-FL vs. other pFL algorithms

with $N = 100$ clients, batch size 128, and learning rate $\eta = 0.01$. When the incentive is disabled clients do not consider maximizing their incentive while sending preference bids. Instead, clients send preference bids with random cluster choices to the scheduler as in FedAvg [46].

Figure 7, shows the comparison with incentives (I) and without (NI) incentives, and the results show a higher accuracy with incentives enabled under a majority of data scenarios suggesting the importance of incentives in pFL.

Table XI shows the test accuracies of the cluster-level models. IP-FL(I) indicates that incentives are enabled and PI-F(NI) shows the accuracies when incentives are disabled. In general IP-FL(I) outperforms IP-FL(NI) in terms of test accuracy for all partitions. The important point to note here is that the incentive mechanism in IP-FL directly motivates clients to join clusters in which they can make the most contribution. This results in accurate clustering based on client data distributions and good-quality personalized models. This is indicated by the performance of IP-FL(NI), i.e without incentive, cluster-level models are unable to dominate a single distribution and only perform well for a single distribution for all partitions except 30:70. Compared to this, in IP-FL(I) each cluster-level model dominates and performs well for their distribution.

J. Convergence speed

Figure 8 shows IP-FL's superior convergence speed and personalized accuracies over other pFL methods, matching the communication rounds per training epoch of clustering-based algorithms [14], [49]. This efficiency stems from incorporating client preferences into post-aggregation updates, avoiding the additional proximal update step needed in clustering-based approaches. Moreover, IP-FL enables optional single-shot personalization with cluster models during training or post-training, further enhancing efficiency.

VII. Cost and Overhead

In terms of computation costs, IP-FL vastly outperforms clustering-based pFL algorithms. The significant advantage arises because other recent cluster-based pFL solutions [49] necessitate an additional update step during training, referred to as proximal update. On the other hand, IP-FL clients only require a single-shot personalization using cluster models, which is not only efficient but also conveniently performed by each client once the training concludes. It is worth noting that this personalization step is not mandatory during training, which further adds to the efficiency of our approach.

VIII. Limitations

Calculating Shapley values for contribution calculation puts some additional overhead on the aggregator server however, we have partially resolved that by using Shapley Values approximations. In the future, we can easily include more lightweight contribution calculation mechanisms as they evolve.

In addition to that, calculating evaluation metrics such as PMA could consume resources, however, we would emphasize that PMA is an evaluation metric employed to demonstrate performance improvements of IP-FL over other pFL algorithms. PMA or threshold calculation is not required for the IP-FL algorithm itself. We have just use it to compare with other pFL algorithms. Therefore, in the practical application of IP-FL, calculating PMA or the threshold is not a requisite step. Even if we take local training as the threshold the results will be the same because the threshold only serves the purpose of comparing the IP-FL algorithm with other pFL algorithms.

IX. Conclusion

In this paper, we proposed IP-FL to address the challenges of incentive provision in pFL for increasing consistent participation by providing appealing personalized models to clients. IP-FL client-centric clustering approach ensures accurate clustering and improved performance even in case of dynamic data distribution shift of the client's local data or inadvertently mistaken clustering decision by the client. Unlike prior works that consider incentivizing and personalization as separate problems, IP-FL solves them as interrelated challenges yielding improvement in pFL performance. Extensive empirical evaluation shows its promising performance compared to other state-of-the-art works.

Acknowledgments

This work is supported in part by the Amazon ML Systems Fellowship, University of Minnesota Grants to Advance Graduate Education (GAGE) Fellowship, Samsung Global Research Outreach (GRO) Award, and NSF grants CSR-2106634 and CSR-2312785.

Authorized licensed use limited to: Florida Atlantic University. Downloaded on September 18,2025 at 15:24:44 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] Haider Ali, Dian Chen, Matthew Harrington, Nathaniel Salazar, Mohannad Al Ameedi, Ahmad Faraz Khan, Ali R. Butt, and Jin-Hee Cho. A survey on attacks and their countermeasures in deep learning: Applications in deep neural networks, federated, transfer, and deep reinforcement learning. *IEEE Access*, 11:120095–120130, 2023.

[2] Rodolfo Stoffel Antunes, Cristiano André da Costa, Arne Küderle, Imrana Abdullahi Yari, and Björn Eskofier. Federated learning for healthcare: Systematic review and architecture proposal. *ACM Trans. Intell. Syst. Technol.*, 13(4), may 2022.

[3] K. S. Arikumar, Sahaya Beni Prathiba, Mamoun Alazab, Thippa Reddy Gadekallu, Sharnil Pandya, Javed Masood Khan, and Rajalakshmi Shenbaga Moorthy. Fl-pmi: Federated learning-based person movement identification through wearable devices in smart healthcare systems. *Sensors*, 22(4), 2022.

[4] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečný, Stefano Mazzocchi, H. Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roselander. Towards federated learning at scale: System design, 2019.

[5] Zheng Chai, Ahsan Ali, Syed Zawad, Stacey Truex, Ali Anwar, Nathalie Baracaldo, Yi Zhou, Heiko Ludwig, Feng Yan, and Yue Cheng. Tifl: A tier-based federated learning system. *CoRR*, abs/2001.09249, 2020.

[6] Huili Chen, Jie Ding, Eric W Tramel, Shuang Wu, Anit Kumar Sahu, Salman Avestimehr, and Tao Zhang. Self-aware personalized federated learning. *Advances in Neural Information Processing Systems*, 35:20675–20688, 2022.

[7] Yae Jee Cho, Divyansh Jhunjhunwala, Tian Li, Virginia Smith, and Gauri Joshi. Maximizing global model appeal in federated learning, 2023.

[8] Liam Collins, Enmao Diao, Tanya Roosta, Jie Ding, and Tao Zhang. Perfedsi: A framework for personalized federated learning with side information. 2022.

[9] Yongheng Deng, Feng Lyu, Ju Ren, Yi-Chao Chen, Peng Yang, Yuezhi Zhou, and Yaoxue Zhang. Fair: Quality-aware federated learning with precise user incentive and model aggregation. In *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, pages 1–10, 2021.

[10] Canh T. Dinh, Nguyen H. Tran, and Tuan Dung Nguyen. Personalized federated learning with moreau envelopes, 2022.

[11] Moming Duan, Duo Liu, Xinyuan Ji, Renping Liu, Liang Liang, Xianzhang Chen, and Yujuan Tan. Fedgroup: Accurate federated learning via decomposed similarity-based clustering. 2021.

[12] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 3557–3568. Curran Associates, Inc., 2020.

[13] Liang Gao, Li Li, Yingwen Chen, Wenli Zheng, ChengZhong Xu, and Ming Xu. Fifl: A fair incentive mechanism for federated learning. In *Proceedings of the 50th International Conference on Parallel Processing*, ICPP '21, New York, NY, USA, 2021. Association for Computing Machinery.

[14] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems*, 33:19586–19597, 2020.

[15] Waris Gill, Ali Anwar, and Muhammad Ali Gulzar. Feddebug: Systematic debugging for federated learning applications. In *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, pages 512–523, 2023.

[16] Waris Gill, Ali Anwar, and Muhammad Ali Gulzar. Feddefender: Backdoor attack defense in federated learning. In *Proceedings of the 1st International Workshop on Dependability and Trustworthiness of Safety-Critical Systems with Machine Learned Components*, SE4SafeML 2023, page 6–9, New York, NY, USA, 2023. Association for Computing Machinery.

[17] Waris Gill, Ali Anwar, and Muhammad Ali Gulzar. TraceFL: Interpretability-Driven Debugging in Federated Learning via Neuron Provenance. In *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*. IEEE, 2025.

[18] Heitor Murilo Gomes, Jesse Read, Albert Bifet, Jean Paul Barddal, and João Gama. Machine learning for streaming data: state of the art, challenges, and opportunities. *SIGKDD Explor. Newsl.*, 21(2):6–22, nov 2019.

[19] Farzin Haddadpour and Mehrdad Mahdavi. On the convergence of local descent methods in federated learning, 2019.

[20] Jingoo Han, Ahmad Faraz Khan, Syed Zawad, Ali Anwar, Nathalie Baracaldo Angel, Yi Zhou, Feng Yan, and Ali R. Butt. Heterogeneity-aware adaptive federated learning scheduling. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 911–920, 2022.

[21] Jingoo Han, Ahmad Faraz Khan, Syed Zawad, Ali Anwar, Nathalie Baracaldo Angel, Yi Zhou, Feng Yan, and Ali R. Butt. Tiff: Tokenized incentive for federated learning. In *2022 IEEE 15th International Conference on Cloud Computing (CLOUD)*, pages 407–416, 2022.

[22] Filip Hanzely and Peter Richtárik. Federated learning of a mixture of global and local models, 2021.

[23] Filip Hanzely, Boxin Zhao, and Mladen Kolar. Personalized federated learning: A unified framework and universal optimization techniques. *ArXiv*, abs/2102.09743, 2021.

[24] Sabaat Haroon, Chris Brown, and Muhammad Ali Gulzar. Desql: Interactive debugging of sql in data-intensive scalable computing. *Proc. ACM Softw. Eng.*, 1(FSE), July 2024.

[25] Miao Hu, Di Wu, Yipeng Zhou, Xu Chen, and Min Chen. Incentive-aware autonomous client participation in federated learning. *IEEE Transactions on Parallel and Distributed Systems*, 33(10):2612–2627, 2022.

[26] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas J. Spanos. Towards efficient data valuation based on the shapley value. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 1167–1176. PMLR, 16–18 Apr 2019.

[27] Peter Kairouz, H Brendan McMahan, Andrew Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Christina Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 12(3-4):1–357, 2019.

[28] Shawal Khalid and Chris Brown. Exploring stakeholder challenges in recruitment for human-centric computing research. In *2024 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 432–438, 2024.

[29] Shawal Khalid, Huayu Liang, and Chris Brown. Exploring user perceptions of crypto signals: An empirical study from social media. In *2024 IEEE International Conference on Blockchain (Blockchain)*, pages 483–490, 2024.

[30] Shawal Khalid, Huayu Liang, and Chris Brown. Decoding the influence: Analyzing the impact of crypto signals on software repositories. In *Proceedings of the IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, January 2025.

[31] Ahmad Faraz Khan, Samuel Fountain, Ahmed M. Abdelmoniem, Ali R. Butt, and Ali Anwar. Flstore: Efficient federated learning storage for non-training workloads, 2025.

[32] Ahmad Faraz Khan, Azal Ahmad Khan, Ahmed M. Abdelmoniem, Samuel Fountain, Ali R. Butt, and Ali Anwar. Float: Federated learning optimizations with automated tuning. In *Proceedings of the Nineteenth European Conference on Computer Systems*, EuroSys '24, page 200–218, New York, NY, USA, 2024. Association for Computing Machinery.

[33] Ahmad Faraz Khan, Yuze Li, Xinran Wang, Sabaat Haroon, Haider Ali, Yue Cheng, Ali R. Butt, and Ali Anwar. Towards cost-effective and resource-aware aggregation at edge for federated learning. In *2023 IEEE International Conference on Big Data (BigData)*, pages 690–699, 2023.

[34] Azal Ahmad Khan, Ahmad Faraz Khan, Haidar Ali, and Ali Anwar. Personalized federated learning techniques: Empirical analysis. In *2024 IEEE International Conference on Big Data (BigData)*, pages 1333–1339, 2024.

[35] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[36] Viraj Kulkarni, Milind Kulkarni, and Aniruddha Pant. Survey of personalization techniques for federated learning. In *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, pages 794–797, 2020.

[37] Fan Lai, Xiangfeng Zhu, Harsha V. Madhyastha, and Mosharaf Chowdhury. Oort: Efficient federated learning via guided participant selection, 2021.

Authorized licensed use limited to: Florida Atlantic University. Downloaded on September 18,2025 at 15:24:44 UTC from IEEE Xplore. Restrictions apply.

[38] Qi Le, Enmao Diao, Xinran Wang, Ali Anwar, Vahid Tarokh, and Jie Ding. Personalized federated recommender systems with private and partially federated autoencoders. *arXiv preprint arXiv:2212.08779*, 2022.

[39] Qi Le, Enmao Diao, Xinran Wang, Ahmad Faraz Khan, Vahid Tarokh, Jie Ding, and Ali Anwar. Dynamicfl: Federated learning with dynamic communication resource allocation. In *2024 IEEE International Conference on Big Data (BigData)*, pages 998–1008, 2024.

[40] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Federated multi-task learning for competing constraints. *CoRR*, abs/2012.04221, 2020.

[41] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks, 2020.

[42] Zelei Liu, Yuanyuan Chen, Han Yu, Yang Liu, and Lizhen Cui. Gtg-shapley: Efficient and accurate participant contribution evaluation in federated learning. *ACM Trans. Intell. Syst. Technol.*, 13(4), may 2022.

[43] Ankur Mallick, Kevin Hsieh, Behnaz Arzani, and Gauri Joshi. Matchmaker: Data drift mitigation in machine learning for large-scale systems. In D. Marculescu, Y. Chi, and C. Wu, editors, *Proceedings of Machine Learning and Systems*, volume 4, pages 77–94, 2022.

[44] Y. Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. Three approaches for personalization with applications to federated learning. *ArXiv*, abs/2002.10619, 2020.

[45] Othmane MARFOQ, Giovanni Neglia, Aurélien Bellet, Laetitia Kameni, and Richard Vidal. Federated multi-task learning under a mixture of distributions. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[46] H. Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. Federated learning of deep networks using model averaging. *CoRR*, abs/1602.05629, 2016.

[47] Libby Plummer. How federated analytics can help banks to share data securely.

[48] Python. Cpython/functions.rst at main | python/cpython. https://github.com/python/cpython/blob/main/Doc/library/functions.rst, 2002.

[49] Yichen Ruan and Carlee Joe-Wong. Fedsoft: Soft clustered federated learning with proximal local updating. In *AAAI*, 2022.

[50] Yuxin Shi, Han Yu, and Cyril Leung. Towards fairness-aware federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–17, 2023.

[51] Zhuan Shi, Lan Zhang, Zhenyu Yao, Lingjuan Lyu, Cen Chen, Li Wang, Junhao Wang, and Xiang-Yang Li. Fedfaim: A model performance-based fair incentive mechanism for federated learning. *IEEE Transactions on Big Data*, pages 1–13, 2022.

[52] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *CoRR*, abs/2103.00710, 2021.

[53] Ming Tang and Vincent W.S. Wong. An incentive mechanism for cross-silo federated learning: A public goods perspective. In *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, pages 1–10, 2021.

[54] Xueyang Tang, Song Guo, and J Guo. Personalized federated learning with clustered generalization. *ArXiv*, abs/2106.13044, 2021.

[55] Xueyang Tang, Song Guo, and Jingcai Guo. Personalized federated learning with contextualized generalization. In Lud De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 2241–2247. International Joint Conferences on Artificial Intelligence Organization, 7 2022. Main Track.

[56] Xinran Wang, Qi Le, Ahmad Faraz Khan, Jie Ding, and Ali Anwar. Icl: An incentivized collaborative learning framework. In *2024 IEEE International Conference on Big Data (BigData)*, pages 94–103, 2024.

[57] Shuyue Wei, Yongxin Tong, Zimu Zhou, and Tianshu Song. *Efficient and Fair Data Valuation for Horizontal Federated Learning*, pages 139–152. 11 2020.

[58] Chenglong Ye, Reza Ghanadan, and Jie Ding. Meta clustering for collaborative learning. *Journal of Computational and Graphical Statistics*, pages 1–10, 2022.

[59] Jianqing Zhang, Yang Hua, Hao Wang, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan. Fedala: Adaptive local aggregation for personalized federated learning, 12 2022.

[60] Jingwen Zhang, Yuezhou Wu, and Rong Pan. Incentive mechanism for horizontal federated learning based on reputation and reverse auction. In *Proceedings of the Web Conference 2021*, WWW '21, page 947–956, New York, NY, USA, 2021. Association for Computing Machinery.

[61] Lefeng Zhang, Tianqing Zhu, Ping Xiong, Wanlei Zhou, and Philip S. Yu. A robust game-theoretical federated learning framework with joint differential privacy. *IEEE Transactions on Knowledge and Data Engineering*, 35(4):3333–3346, 2023.

[62] Michael Zhang, Karan Sapra, Sanja Fidler, Serena Yeung, and Jose M. Alvarez. Personalized federated learning with first order model optimization, 2021.

[63] Zirui Zhou, Lingyang Chu, Changxin Liu, Lanjun Wang, Jian Pei, and Yong Zhang. Towards fair federated learning. In *Proceedings of the 27th ACM SIGKDD*, KDD '21, page 4100–4101, New York, NY, USA, 2021. Association for Computing Machinery.

Authorized licensed use limited to: Florida Atlantic University. Downloaded on September 18,2025 at 15:24:44 UTC from IEEE Xplore. Restrictions apply.