FedClusterEnsemble: Enhancing Clustered Federated Learning Through Confidence-Based Ensembles and Cyclic Intra Cluster Client Selection

Artur Sousa Freitas[a,*], Ademar Takeo Akabane[b], Júlio Cezar Estrella[a]

[a]*Institute of Mathematics and Computer Sciences, University of São Paulo, São Carlos, Brazil*
[b]*Pontifical Catholic University of Campinas, Polytechnique School, Postgraduate Program in Urban Infrastructure Systems, Campinas, Brazil*

Abstract

Federated Learning (FL) has emerged as a transformative paradigm for decentralized model training, enabling multiple clients to collaboratively learn without sharing sensitive data. However, FL's performance is often hindered by non-independent and identically distributed (non-IID) data across clients, resulting in substantial degradation in model accuracy. Clustered Federated Learning (CFL) addresses this challenge by grouping clients based on data similarity to improve training efficiency. Despite these advancements, existing CFL frameworks lack effective mechanisms to support inference-only clients—those that do not participate in training but require reliable model predictions. This paper introduces a novel ensemble-based strategy tailored for inference-only clients with highly skewed data distributions. Additionally, we propose a cyclic client selection strategy that significantly reduces communication overhead while maintaining model performance. Our framework constructs multiple client clusters based on data similarity, allowing each cluster to train models independently while adapting to diverse data distributions through hyperparameter tuning. During inference, a confidence-based ensemble method is employed, selecting the model with the highest predictive confidence for each test sample instead of conventional averaging

*Corresponding author: Artur Sousa Freitas
Email addresses: arturf@usp.br (Artur Sousa Freitas), ademar.akabane@puc-campinas.edu.br (Ademar Takeo Akabane), julio.estrella@usp.br (Júlio Cezar Estrella)

or majority voting. This approach enhances predictive accuracy and ensures robust performance even when inference-only clients encounter previously unseen data distributions. Experiments conducted on benchmark datasets, including CIFAR-10 and SVHN, demonstrate that the proposed method outperforms FedAvg in both accuracy and communication efficiency, particularly in federated settings characterized by extreme data heterogeneity. Moreover, the framework is especially suitable for IoT applications—such as edge computing, sensor networks, and vehicular systems—where data heterogeneity and limited communication resources are critical challenges. These contributions advance the practical deployment of FL in real-world domains such as healthcare, IoT, and vehicular networks.

Keywords: Federated Learning, Intra Cluster Client Selection, Non-IID Data, Edge Computing

---

1. Introduction

The proliferation of distributed data sources across diverse domains, such as healthcare, the Internet of Things (IoT), and vehicular networks, has led to an increasing demand for collaborative machine learning frameworks that preserve data privacy. Federated Learning (FL) has emerged as a transformative paradigm in decentralized machine learning, allowing multiple clients to collaboratively train a shared model without exposing their raw data, thereby addressing critical privacy and security concerns [**?** **?** **?**]. By ensuring that raw data remains on local devices, FL enhances privacy and compliance with data protection regulations such as the General Data Protection Regulation (GDPR) and the Health Insurance Portability and Accountability Act (HIPAA). FL has been successfully applied in privacy-sensitive domains, including healthcare [**?**], Internet of Things (IoT) networks [**?**], and autonomous vehicle systems [**?**], where data is inherently distributed and sensitive [**?**]. In healthcare, FL facilitates collaborative model training across hospitals while preserving patient confidentiality. In IoT networks, the heterogeneity of edge devices and the presence of non-independent and identically distributed (non-IID) data pose unique challenges that require efficient model aggregation strategies. Similarly, in autonomous vehicle networks, the dynamic network environments, such as the mobility of devices and the randomness of link connections, further make sensory data not only heterogeneous but also non-stationary [**?**]. In such scenarios, FL must

accommodate real-time constraints, adapt to highly dynamic data distributions, and ensure robust and reliable predictions.

Despite its advantages, FL faces several challenges that hinder its deployment in real-world scenarios. A fundamental issue is the presence of non-IID data across clients, which arises when data distributions vary significantly due to differences in device characteristics, user behavior, or environmental conditions. For instance, IoT sensors capture distinct environmental patterns, and in healthcare, patient demographics and medical histories introduce substantial variability. This heterogeneity disrupts the convergence of global models, leading to biased updates that degrade performance and generalization capabilities [**?** **?**]. The presence of non-IID data also results in increased communication overhead, imbalanced class distributions, and uneven local model updates, further impacting FL convergence and performance [**?**]. Conventional FL algorithms, such as Federated Averaging (FedAvg) [**?**] and its extensions, often struggle under non-IID conditions, necessitating more sophisticated approaches to ensure model robustness.

Communication constraints further exacerbate these challenges, particularly in resource-limited environments such as IoT and embedded systems. FL inherently relies on iterative model updates exchanged between clients and a central server, imposing significant bandwidth requirements. In real-world deployments, where network connectivity may be intermittent or low-bandwidth, reducing communication overhead is critical. Recent studies, such as FedArtML [**?**], have explored strategies to optimize communication efficiency, yet the trade-off between update frequency and model performance remains an open research question.

Another key challenge lies in enabling robust inference mechanisms that generalize well across diverse and unseen data distributions. Real-world deployments frequently encounter inference-only clients where training data from new clients is unavailable. For example, in vehicular networks, models must adapt to data from new vehicles entering the system without retraining. Similarly, in healthcare applications, models must provide reliable predictions for patients whose data distributions differ from those seen during training [**?**]. Ensuring robust inference in such scenarios is critical for the success of FL in practical applications.

To address these challenges, this work proposes an enhanced clustered federated learning (CFL) framework tailored for real-world deployments. The proposed approach clusters clients based on data similarity, enabling specialization within groups to address non-IID distributions. Additionally,

a confidence-based ensemble inference mechanism is introduced to enhance prediction accuracy and robustness for unseen and inference-only clients. To mitigate communication costs, the framework incorporates a cyclic client selection strategy, promoting fairness while reducing the frequency of updates.

The key contributions of this work are summarized as follows:

- **Cluster-Based Training:** A novel CFL framework that clusters clients based on data similarity, addressing the challenges posed by non-IID data distributions;
- **Confidence-Based Ensemble Inference:** An inference mechanism that enhances prediction robustness and accuracy, particularly for unseen and inference-only clients;
- **Communication Efficiency:** A cyclic client selection strategy that optimizes communication overhead while maintaining fairness and performance.

While existing Clustered FL approaches tackle individual aspects of these challenges, a unified framework that simultaneously addresses non-IID data distributions, inference generalization, and communication efficiency remains lacking. Our approach uniquely integrates cyclical client selection into a trust-based ensemble framework, ensuring robust inference while minimizing communication overhead.

The remainder of this paper is organized as follows: Section 2 provides background on Federated Learning, highlighting its main challenges and reviewing recent advancements in the field. Additionally, in Section 3 we discuss related work and position our approach in comparison to existing methods. In Section 4 we introduce our proposed solution, detailing its key components and how it addresses the identified challenges. Section 5 presents our experimental evaluation on two benchmark datasets, CIFAR-10 and SVHN, assessing both predictive performance and communication efficiency while comparing our approach against the state-of-the-art FedAvg algorithm. Finally, Section 6 summarizes our contributions, and Section 7 outlines potential directions for future work.

2. Background

Federated Learning (FL) is a decentralized approach to training machine learning models collaboratively across multiple clients while preserving data

privacy. Unlike traditional centralized training, FL allows clients to train local models on their private data and share only model updates with a central server, ensuring that sensitive data remains localized. This paradigm has gained significant attention due to its applications in privacy-critical domains such as healthcare, IoT, and autonomous systems [**?**]. Formally, consider a system with K clients, each possessing a local dataset D_k. The objective of FL is to optimize a global model w by minimizing the following empirical risk function [**?**], Equation 1:

$$\min_{w} : F(w) = \sum_{k=1}^{K} p_k F_k(w), \tag{1}$$

where $F_k(w)$ represents the local objective function for client k, defined as:

$$F_k(w) = \mathbb{E}_{(x,y)\sim D_k}[\ell(w; x, y)], \tag{2}$$

where $l(w; x, y)$ is the loss function evaluated on the local dataset D_k, and p_k denotes the weight assigned to client k, often based on the proportion of local data.

A seminal contribution to FL is Federated Averaging (FedAvg) [**?**], which introduced a straightforward aggregation mechanism that iteratively combines locally trained models from clients to update a global model. The algorithm selects a subset of clients in each communication round, who then train their models locally using their private data before reverting updates to the server. These updates are aggregated to refine the global model. Algorithm 1 outlines the main steps of FedAvg.

Algorithm 1 Federated Averaging (FedAvg)

Input: Number of communication rounds T, number of clients K, local epochs E, learning rate η.
Initialize: Server initializes global model w_0.
for $t = 0, \ldots, T-1$ **do**
    Server selects a subset of clients $S_t \subseteq \{1, \ldots, K\}$.
    for each client $k \in S_t$ **in parallel do**
        Client k initializes $w_k \leftarrow w_t$.
        for each local epoch $e = 1, \ldots, E$ **do**
            Client k updates w_k using stochastic gradient descent:
            $w_k \leftarrow w_k - \eta \nabla \ell(w_k; D_k)$ ▷ Gradient update
        end for
        Client k sends updated model w_k to the server.
    end for
    Server aggregates the updates:
    $w_{t+1} \leftarrow \sum_{k \in S_t} \frac{n_k}{\sum_{j \in S_t} n_j} w_k$ ▷ Weighted aggregation
end for
Output: Final global model w_T.

3. Related Works

FedAvg achieved remarkable success in IID scenarios due to its simplicity and scalability. However, its performance under non-IID conditions was significantly hindered by client drift, a phenomenon where local model updates diverge from the global objective. This limitation sparked efforts to develop methods that explicitly address client drift and its associated challenges.

Among these efforts, FedProx [**?**] introduced a proximal term in the optimization objective to penalize updates that deviate significantly from the global model, thereby mitigating client drift and improving convergence stability. Another notable approach, SCAFFOLD [**?**], addressed this issue through variance reduction techniques, which stabilized updates by explicitly correcting client-side optimization dynamics. While these methods demonstrated improved training convergence, they often left gaps in robustness during inference, especially when clients encountered data distributions that were not represented during training.

The inherent heterogeneity of client data has also motivated research into cluster-based FL approaches, which partition clients into groups based

on data similarity. By creating clusters, these methods enable the training of specialized models that cater to the unique characteristics of each group. CADIS [**?**], for example, employed penultimate-layer similarity to group clients and used knowledge distillation to enhance generalization and reduce overfitting. This approach improved accuracy by up to 16% compared to FedAvg in scenarios with significant data heterogeneity. Similarly, IFCA [**?**] proposed a dynamic federated clustering algorithm that adaptively assigned clients to clusters during training, tailoring models to the specific data distributions within each cluster. While effective, these methods face limitations during inference, particularly for clients whose data do not align with any of the trained clusters.

Efforts to enhance the robustness of FL systems have also focused on calibration and adaptation techniques, which aim to align local client updates with the global model objectives more effectively. For instance, LEDA-FL [**?**] utilized Conditional Variational AutoEncoders (CVAEs) to align label-wise feature distributions among clients. By reducing the divergence between local and global updates, LEDA-FL achieved a 6.2% accuracy improvement on the CIFAR-100 dataset while reducing communication costs by 60%. Complementing this, FedFusion [**?**] introduced a data-agnostic distribution fusion approach, employing Variational AutoEncoders to infer global data distributions and optimize model aggregation. These methods marked significant advancements in handling non-IID data but often required assumptions about client data distributions that may not hold in real-world scenarios.

The challenge of communication efficiency has also been a critical area of research, as frequent model updates impose substantial bandwidth requirements, especially in large-scale FL deployments. Techniques such as rFedAvg+ [**?**] introduced distribution regularization to reduce dataset discrepancies among clients, effectively minimizing the number of communication rounds needed for convergence. Furthermore, tools like FedArtML [**?**] have facilitated controlled dataset partitioning into non-IID configurations, enabling systematic experimentation and optimization. These tools demonstrated their utility by achieving robust performance with only a 2% F1-Score difference between simulated and real-world datasets, underscoring their value in advancing communication-efficient FL systems.

Personalized FL approaches have emerged as another promising avenue, addressing the critical need for robust inference in scenarios where clients operate under unique or unseen data distributions. Mixed Data Calibration (MIDAC) [**?**] tackled individual data biases by creating synthetic

mixed data points for calibration, improving accuracy on the CIFAR-10 and BloodMNIST datasets while maintaining low computational overhead and strong privacy guarantees. Similarly, FedDDC [**?**] addressed long-tailed and non-IID data distributions by decoupling feature extraction and classification tasks, reweighting client contributions, and incorporating knowledge distillation. This approach achieved up to a 60% improvement in accuracy compared to traditional baselines, demonstrating robustness across a wide range of non-IID scenarios.

Despite these advancements, significant challenges remain, particularly in ensuring robust inference for clients with data distributions entirely unseen during training. Existing approaches, such as CADIS and LEDA-FL, have demonstrated considerable success in improving accuracy for non-IID settings. However, these methods lack mechanisms for dynamically selecting the optimal model during inference, which is crucial for inference-only clients. To address this gap, our proposed framework introduces a confidence-based ensemble strategy that selects the most suitable cluster model for each test instance. By leveraging predictive confidence, this approach ensures accurate predictions even in scenarios involving unseen distributions.

Additionally, the integration of cyclic client selection mechanisms in our framework enhances communication efficiency by promoting fairness and diversity in client participation. This reduces redundant updates without compromising model performance, further contributing to the scalability and applicability of FL in diverse settings. By combining cluster-based training, adaptive client selection, and confidence-based ensemble inference, our framework provides a holistic solution to the intertwined challenges of data heterogeneity, communication efficiency, and robust inference.

This work advances the state of the art in FL, paving the way for its broader applicability in real-world heterogeneous environments, where non-IID data distributions and unseen inference scenarios are the norm. By addressing these critical challenges, our contributions lay a foundation for developing robust and efficient FL systems capable of operating effectively across diverse domains and use cases, see Table 1.

Table 1: Comparison of Federated Learning Methods

Method	**Non-IID Handling**	**Communication Efficiency**	**Highly Skewed Non-IID Handling**
FedAvg	×	×	×
FedProx	✓	×	×
SCAFFOLD	✓	×	×
CADIS	✓	×	×
IFCA	✓	×	×
LEDA-FL	✓	✓	×
FedFusion	✓	✓	×
MIDAC	✓	✓	×
FedDDC	✓	✓	×
FedClusterEnsemble	✓	✓	✓

3.1. Comparisons with Existing Approaches

In this section, we compare our proposed approach, called **FedClusterEnsemble**, with recent solutions in Federated Learning (FL). The goal is to emphasize how our method fills gaps related to: (i) clustering of highly non-IID clients, (ii) a confidence-based ensemble mechanism, (iii) cyclic client selection (round-robin) to reduce communication overhead, and (iv) support for inference-only clients.

Some works, such as Li et al. [**?**] and Hsu et al. [**?**], discuss the impact of non-IID data on convergence and present empirical analyses, respectively, but neither explores an ensemble mechanism nor a specific cyclic selection strategy. Karimireddy et al. [**?**] and Wang et al. [**?**] focus on mitigating client-drift or handling local objective inconsistencies, yet they do not address inference in clients that do not participate in training.

On the other hand, Qi et al. [**?**] propose an active client selection strategy to improve communication efficiency, although they do not include a confidence-based ensemble. In [**?**], the idea of personalization layers helps adapt to heterogeneous data distributions, but it does not employ clustering. Shil et al. [**?**] incorporate ensembling in FL, yet they do not include parameter-based client grouping or cyclic selection. Lian et al. [**?**] focus on domain-invariant learning and decentralized SGD, respectively, without providing explicit support for isolated inference in new clients. Yu et al. [**?**] offers a broad review of FL in edge computing scenarios, but it does not introduce a solution that jointly covers clustering, confidence-based ensemble, and cyclic selection.

By contrast, **FedClusterEnsemble** addresses these gaps by:

- Grouping clients via parameter similarity (DBSCAN),
- Employing a confidence-based ensemble to handle extremely heterogeneous data,
- Using cyclic selection to reduce overhead and maintain fairness, and
- Supporting inference-only clients who do not contribute local training.

Table 2 provides a succinct overview, using a two-column format, showing how each method deals (or not) with non-IID data, client selection, ensemble methods, and inference-only support.

Table 2: Comparisons with Existing Approaches in Federated Learning

Reference	**Main Focus**	**Non-IID**	**Selection / Communication**	**Clustering / Ensemble**	**Inference-Only**
Li et al. [**?**]	FedAvg convergence	Theoretical	Fixed frequency	No / No	No
Hsu et al. [**?**]	Empirical non-IID study	Yes	Not highlighted	No / No	No
Karimireddy et al. [**?**]	Reducing client-drift	Yes	Partial optimization	No / No	No
Wang et al. [**?**]	Objective inconsistency	Yes	Fewer rounds	No / No	No
Haddadpour et al. [**?**]	Personalization layers	Heterogeneous data	Not prioritized	No / No	No
Qi et al. [**?**]	Active selection (FedSampler)	Yes	Optimized overhead	No / No	No
Li et al. [**?**]	Ensembling (FedEM)	Moderate	Does not address cyclic selection	No / Yes	No
Lian et al. [**?**]	Decentralized / parallel SGD	Generic	Distributed approach	No / No	No
Xia et al. [**?**]	Comprehensive survey on FL (edge)	Covers non-IID	Conceptual	No / No	No
This Work	**Clustering + Confidence Ensemble + Cyclic Selection**	**Yes (highly heterogeneous)**	**Reduces overhead, fairness**	**Yes / Yes (confidence-based)**	**Yes (inference-only)**

As shown, some approaches address specific FL challenges such as communication overhead or heterogeneity, while others introduce ensemble methods. However, few methods combine *all these aspects* in a single solution. **FedClusterEnsemble** moves forward by offering:

1. Specialized models for diverse client clusters,
2. A confidence-based ensemble to manage heterogeneous environments,
3. More efficient communication (cyclic selection), and
4. Direct support for inference-only devices.

Hence, this work contributes by simultaneously tackling key challenges that

are often treated separately in the literature, providing a practical and scalable solution for FL scenarios where data heterogeneity and communication constraints are critical.

4. Proposed Solution

To address the limitations discussed in the previous section and enhance the performance of clustered federated learning under non-IID conditions, we introduce a novel federated learning framework that clusters clients based on data similarity and leverages an ensemble inference mechanism to improve centralized prediction accuracy. The proposed approach aims to optimize the performance of the model in heterogeneous environments by tailoring the training process to the specific characteristics of each group of clients. In the following, we outline the key steps of our framework.

1. **Initial Training Round:**
   - **Client Selection:** In the first round, **all clients** are selected to participate in training, ensuring that the model is exposed to a wide range of data distributions from the outset.
   - **Local Training:** Each client trains the global model on its local dataset and sends the updated model parameters to the central server.
2. **Model Parameter Extraction and Clustering:**
   - **Feature Extraction:** Upon receiving the models, the server extracts the **last-layer parameters** from each client's model. These parameters capture high-level features learned by the model.
   - **Clustering with DBSCAN:** The server applies the **DBSCAN** clustering algorithm to the extracted last-layer parameters. The goal is to group clients with similar data distributions into clusters based on the similarity of their learned representations. This clustering is performed only during the first round, as the data is **stationary**, and client distributions are assumed not to change over time.
3. **Cluster-Based Model Aggregation:**
   - **Separate Aggregation:** Once the clusters are identified, the server performs **independent model aggregation** within each

cluster. Each cluster's clients contribute to a separate aggregated model that better captures the nuances of their shared distribution.

- **Model Distribution:** The server then distributes the **cluster-specific models** back to the clients in each respective cluster, allowing them to fine-tune models that are specialized for their data.

4. **Cyclic Client Selection in Subsequent Rounds:**
   - **Client Selection Rotation:** In subsequent rounds, clients are selected using a **round-robin strategy** within each cluster, ensuring periodic participation while reducing communication costs.
   - **Selection Logic:** Instead of selecting all clients in every round, a **subset of clients** (e.g., 70%) is chosen in a cyclic manner. Clients are rotated following a **deterministic schedule**, ensuring each client contributes to training at regular intervals.
   - **Cluster Interactions:** Clients remain assigned to their respective clusters, and at each round, a **different subset of clients** from each cluster is selected to participate. This ensures all clusters contribute to updates while maintaining efficiency.
   - **Fairness Considerations:** The cyclic selection guarantees that no client is persistently excluded, ensuring balanced participation and a fair training process across rounds.
5. **Ensemble Inference with Confidence-Based Selection:**
   - **Ensemble Formation:** At the beginning of each round after clustering, the aggregated models from each cluster are stored in a list, forming an **ensemble**.
   - **Inference Process:** During inference, input samples are passed through each model in the ensemble.
   - **Logit Collection:** Each model in the ensemble outputs a vector of **logits** (the raw scores before applying softmax) for the input data.
   - **Prediction Selection:** For each sample, the final class label is determined by selecting the **class corresponding to the highest logit value** across all models in the ensemble. This ensures

that the model with the highest confidence for a given input provides the prediction.

6. **Training in Subsequent Rounds:**
   - **Continued Training:** In subsequent rounds, clients continue training using their cluster-specific models.
   - **Model Aggregation:** After each round, the server continues to aggregate models within each cluster, refining them over time.
   - **Ensemble Updating:** The ensemble is updated at each round with the latest aggregated models, ensuring that the inference process benefits from the most recent training iterations.

Figure 1 illustrates the proposed Clustered Federated Learning (CFL) framework. At the top, a cloud server manages both training and inference.The training process is described in Algorithm 2, where the server organizes training rounds, applies clustering, and maintains separate models for each identified group of clients. After the clustering phase, the server cyclically selects one client per cluster to contribute to the corresponding cluster model. The selected client receives the latest cluster-specific model, performs local training, and transmits updated model parameters back to the server. The server then aggregates these updates separately for each cluster, ensuring models remain specialized for their respective client distributions. Inference-only clients, represented by embedded devices, interact with the server to request predictions. When a request is received, the ensemble-based mechanism is triggered, forwarding the input data to all cluster models. Each model outputs a logit vector, and the final prediction is determined by selecting the class with the highest confidence across all models. This process is described in Algorithm 3, which details how the ensemble inference mechanism ensures that the most confident model contributes to the final decision.

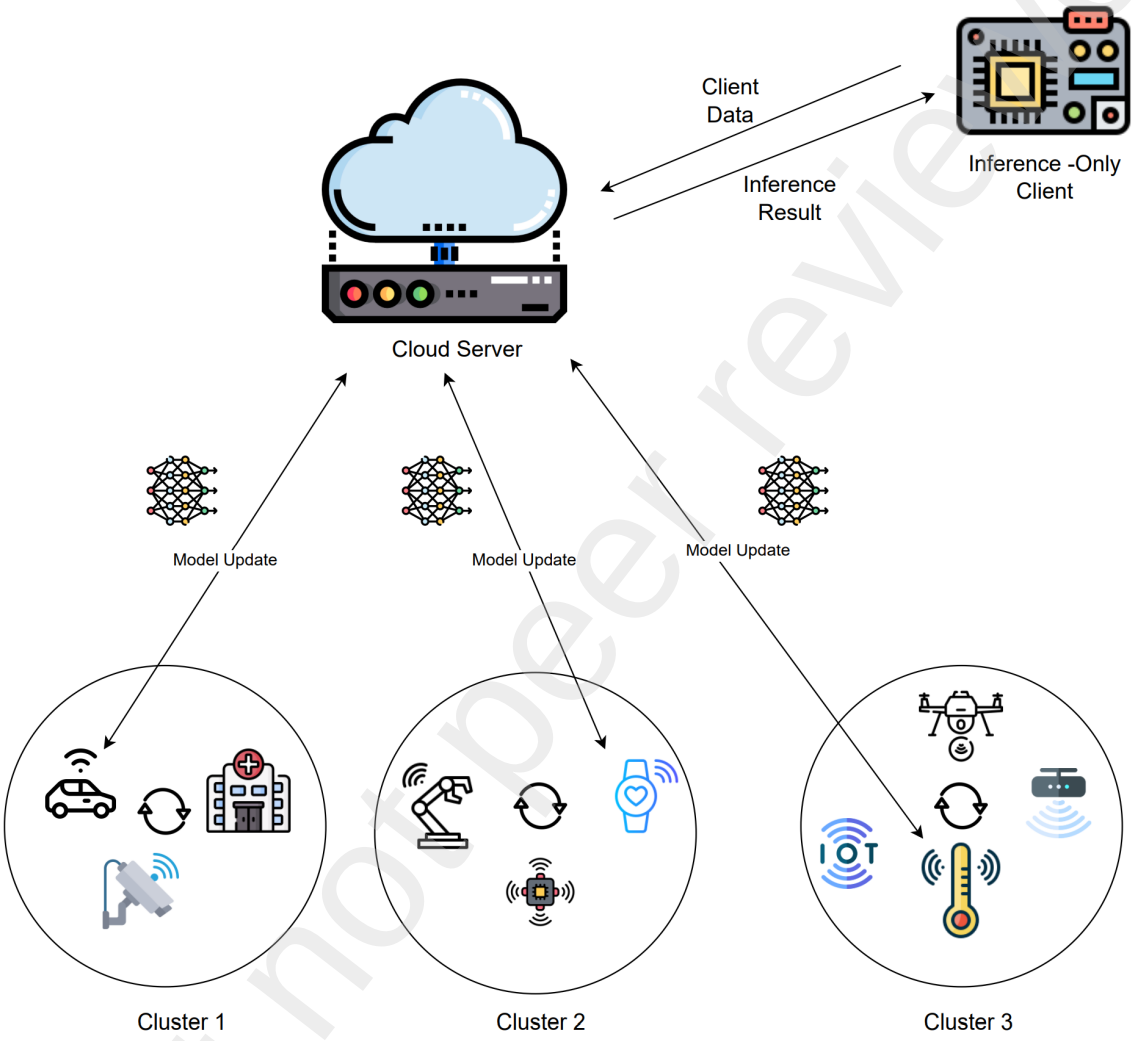

Figure 1: FedClusterEnsemble high-level architecture.

Algorithm 2 Clustered Federated Learning Training

Input: Number of rounds T, client set K, initial global model w_0, local epochs E, learning rate η.
Output: Cluster-specific models $w_T^{(m)} m = 1^M$.
Server initializes global model w_0.
Round 1: Global Training
All clients $k \in K$ train locally using w_0 for E epochs.
Clients send local models w_k to server.
Server aggregates updates from all clients:
$w_1 \leftarrow \sum k \in K \frac{n_k}{\sum_{j \in K} n_j} w_k$
Server extracts last-layer parameters from w_1.
Server applies DBSCAN clustering to parameters, forming clusters $C_1, C_2, \ldots, C_M$.
for each cluster C_m **do**
    Server initializes and maintains a separate model for each cluster.
    Aggregate models independently:
    $w_m^{(1)} \leftarrow \sum_{k \in C_m} \frac{n_k}{\sum_{j \in C_m} n_j} w_k$
end for
for round $t = 2, \ldots, T$ **do**
    for each cluster C_m **do**
        Select one client from C_m in a cyclic manner.
        Client performs local training on cluster model $w_m^{(t-1)}$.
        Server aggregates the updated model:
        $w_m^{(t)} \leftarrow w_k$
    end for
end for

Algorithm 3 Ensemble Inference with Confidence-Based Selection

Input: Ensemble models $w_m m = 1^M$, input sample x.
Output: Prediction $ypred(x)$.
for each model w_m in ensemble **do**
    Compute model logits:
    $l_m(x) \leftarrow w_m(x)$
end for
Select model with highest confidence:
$m^* \leftarrow \arg\max_m \max(l_m(x))$
Determine final prediction:
$y_{pred}(x) \leftarrow \arg\max(l_{m^*}(x))$

In summary, the proposed approach clusters clients based on their learned representations in the first round and continues refining models within each cluster throughout subsequent rounds. The method ensures efficient communication while maintaining fairness across rounds by incorporating a **cyclic client selection strategy**. Furthermore, using **confidence-based ensemble inference** allows the system to leverage the most reliable model predictions dynamically.

A key advantage of this framework is its ability to handle newly arriving **inference-only clients**—clients that were not present during the initial clustering phase. These clients can seamlessly integrate into the system without requiring explicit retraining or re-clustering. Instead of being assigned to a predefined cluster, an **unseen client** directly queries the **ensemble model**, leveraging the cluster-specific models to obtain predictions. The final output is determined based on the highest confidence prediction among all models in the ensemble. This mechanism enables efficient adaptation to dynamic environments where new clients may request inference without contributing to the training process.

The next section presents experimental results demonstrating the effectiveness of this approach in handling non-IID data, improving accuracy and generalization, and efficiently managing inference requests from unknown clients.

5. Performance Evaluation

In this section, we evaluate the performance of our proposed Clustered Federated Learning (FedClusterEnsemble) strategy in comparison to the tra-

ditional Federated Averaging (FedAvg) method. The evaluation is carried out on two benchmark datasets - CIFAR-10[1] and SVHN[2] — under highly skewed Non-IID data settings, designed to simulate real-world federated learning scenarios where client data distributions are heterogeneous.

We assess the performance of FedClusterEnsemble using two primary metrics:

1. **Centralized Accuracy**: Equation 3 measures the proportion of correctly classified samples on a centralized, balanced test dataset maintained on the server. It reflects the model's generalization performance after federated training. Formally, it is defined as:

$$A_c = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(\hat{y}_i = y_i), \tag{3}$$

   where n is the total number of test samples, y_i is the true label of the i-th sample, and $\hat{y}_i$ is the corresponding predicted label. The indicator function $\mathbb{I}(\hat{y}_i = y_i)$ returns 1 if the prediction is correct and 0 otherwise.
2. **Communication Cost**: The communication cost in this work is defined as the cumulative number of client updates sent to the server throughout training. For each round, the communication cost, Equation 4, is proportional to the number of clients selected to send updates. Therefore, the total communication cost after T rounds is given by:

$$\text{Communication Cost} = \sum_{t=1}^{T} N_t \tag{4}$$

   where T is the total number of rounds and N_t is the number of clients participating in round t.

This metric evaluates the efficiency of the communication process, which is critical in federated learning environments with limited bandwidth.

5.1. Experimental Setup

We conducted our experiments using the **Flower framework [?]**, which provides a flexible and scalable environment for federated learning research. Below, we outline the key components of our experimental setup:

[1]https://www.cs.toronto.edu/ kriz/cifar.html
[2]http://ufldl.stanford.edu/housenumbers/

- **Datasets**:
  - **CIFAR-10**: A dataset consisting of 60,000 32×32 RGB images across 10 classes.
  - **SVHN (Street View House Numbers)**: A dataset with 600,000 32×32 RGB images of digits (0–9).
- **Data Partitioning**: The datasets are partitioned among **n** clients, where $n \in \{10, 20, 30\}$, in a highly skewed **Non-IID** manner to simulate real-world federated learning scenarios with heterogeneous data distributions. The partitioning strategy ensures that each client receives data from a limited subset of classes. Specifically:
  - **Label Groups**: The class labels are divided into three distinct groups:
    - **Group 0**: Classes 0, 1, 2
    - **Group 1**: Classes 3, 4, 5
    - **Group 2**: Classes 6, 7, 8, 9
- **Client Assignment**: Clients are assigned to these label groups in a balanced manner:
  - For $n = 10$: Approximately 3–4 clients per group.
  - For $n = 20$: Approximately 6–7 clients per group.
  - For $n = 30$: Approximately 9–10 clients per group.

  Any remaining clients are evenly distributed among the groups to ensure fair assignment.
- **Data Distribution**: Within each group, the dataset is partitioned so that each client receives data only from the specific classes within their assigned group. This method ensures that each client's dataset is significantly different from the global distribution, reflecting the challenges of Non-IID data in federated learning.
- **Training Configuration**: The table below shows the simulation parameters used in the experiments.
- **Hardware and Framework**: Experiments were run on an **Intel i7 CPU**, **NVIDIA GTX 1080 GPU**, and **32GB RAM** using the **Flower framework** with **PyTorch 1.11** and **TensorFlow 2.7**.

5.2. Results and Analysis

In this subsection, we present the experimental results on the **CIFAR-10** dataset, comparing the performance of our proposed *FedClusterEnsemble* strategy with the baseline *FedAvg* method. The evaluation focuses on two key metrics: **centralized accuracy** and **cumulative accuracy gain (CAG)**. The results are analyzed for different client configurations ($n = 10, 20, 30$) to demonstrate the scalability and effectiveness of our approach under highly skewed Non-IID data distributions.

Centralized Accuracy. We evaluate the centralized accuracy on a **balanced test dataset maintained on the server, which emulates inference-only clients**. This dataset is not used for training and serves as a proxy for assessing the generalization capability of models trained in a federated setting. The results are presented separately for the **CIFAR-10** and **SVHN** datasets to highlight the effectiveness of *FedClusterEnsemble* in handling Non-IID data distributions.

CIFAR-10 Results This subsection compares the proposed and traditional approach (FedAvg) in terms of centralized accuracy throughout the rounds plotted with a confience interval of 95%.

Figures 2, 3, and 4 present a comprehensive evaluation of the proposed FedClusterEnsemble strategy compared to the baseline FedAvg method across 60 training rounds under different client configurations (10, 20, and 30 clients) using the CIFAR-10 dataset. The results clearly demonstrate that FedClusterEnsemble consistently outperforms FedAvg in terms of centralized accuracy.

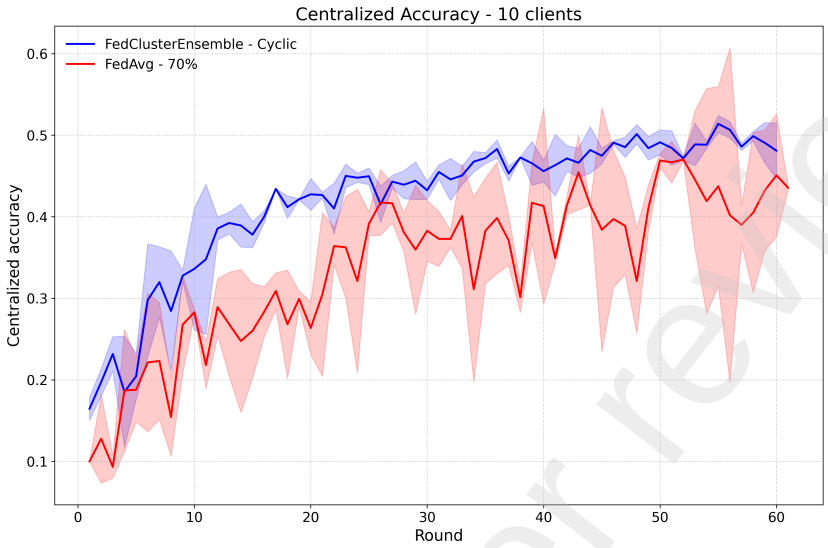

Figure 2: Centralized Accuracy Comparison of Federated Learning Strategies with 10 Clients on the CIFAR-10 Dataset

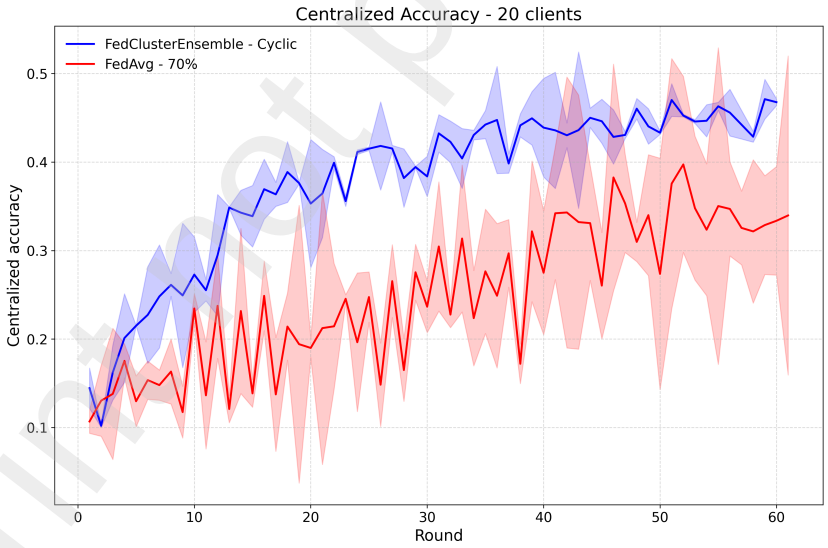

Figure 3: Centralized Accuracy Comparison of Federated Learning Strategies with 20 Clients on the CIFAR-10 Dataset

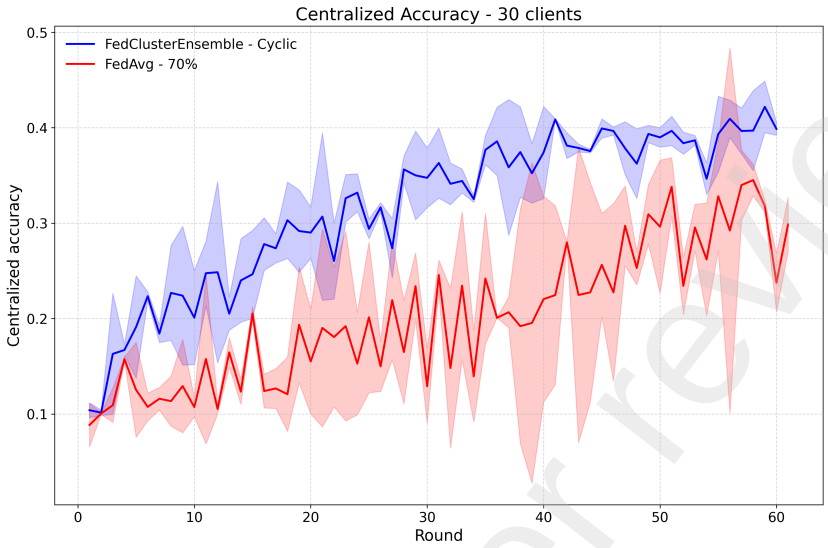

Figure 4: Centralized Accuracy Comparison of Federated Learning Strategies with 30 Clients on the CIFAR-10 Dataset

In the 10-client configuration, FedClusterEnsemble achieves an average accuracy improvement of approximately 6.5 over FedAvg. Similarly, in the 20-client and 30-client configurations, the method achieves accuracy improvements of 5.8 and 7.2

The accuracy curves across different configurations indicate that FedClusterEnsemble achieves faster convergence compared to FedAvg. The proposed strategy maintains superior centralized accuracy throughout the training rounds, demonstrating its robustness across different client configurations.

Overall, the results on CIFAR-10 highlight the practical effectiveness of FedClusterEnsemble in improving centralized accuracy in federated learning scenarios characterized by non-IID client data. The method's ability to achieve higher accuracy compared to baseline approaches makes it a promising solution for applications in edge computing, IoT networks, and privacy-preserving systems.

SVHN results Street View House Numbers (SVHN) is a digit classification benchmark dataset that contains 600,000 32×32 RGB images of printed digits (from 0 to 9) cropped from pictures of house number plates. The cropped images are centered in the digit of interest, but nearby digits and other distractors are kept in the image. SVHN has three sets: the train-

ing, testing sets, and an extra set with 530,000 images that are less difficult and can be used for helping with the training process.

Figures 4, 5, and 6 present the centralized accuracy evaluation of the proposed FedClusterEnsemble strategy compared to the baseline FedAvg method across 60 training rounds under different client configurations (10, 20, and 30 clients) using the SVHN dataset. Unlike the CIFAR-10 results, FedAvg consistently outperforms FedClusterEnsemble on the SVHN dataset across all configurations.

In the 10-client configuration (Figure 5), FedAvg achieves higher centralized accuracy throughout the training rounds. The performance gap becomes noticeable early in the training process and remains consistent across the rounds. However, it is worth noting that FedClusterEnsemble shows more stable behavior, with narrower confidence intervals around the accuracy curves, indicating reduced variability across rounds.

In the 20-client configuration (Figure 6), FedAvg continues to maintain a significant accuracy advantage over FedClusterEnsemble. Despite this, FedClusterEnsemble demonstrates more stable performance across rounds, which could be beneficial in scenarios where consistency in predictions is prioritized.

In the 30-client configuration (Figure 7), FedAvg further solidifies its superior performance over FedClusterEnsemble. The accuracy gap remains consistent, but once again, FedClusterEnsemble exhibits more stable behavior with reduced performance variability.

Interestingly, while FedAvg demonstrates better performance in terms of centralized accuracy on SVHN, FedClusterEnsemble achieves competitive accuracy levels across different client configurations. These observations suggest that FedClusterEnsemble could still be a viable option in scenarios where stability and reduced variability are prioritized in federated learning deployments.

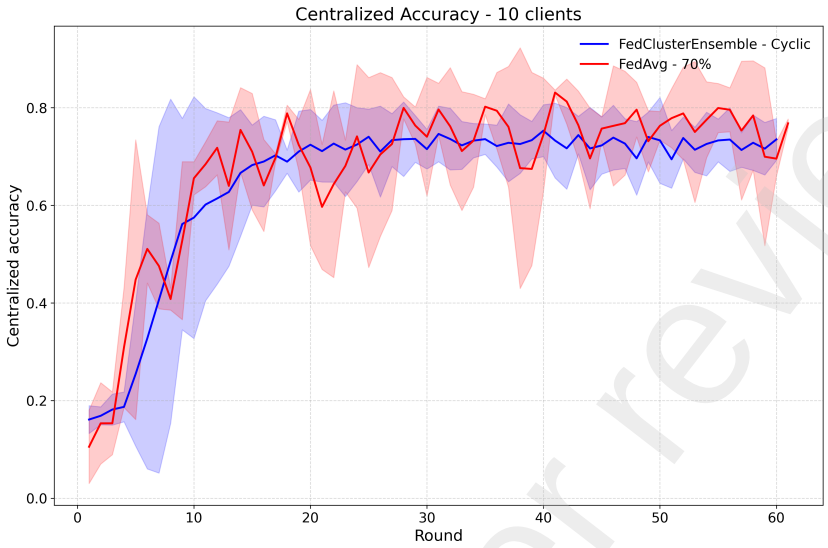

Figure 5: Centralized Accuracy Comparison of Federated Learning Strategies with 10 Clients on the CIFAR-10 Dataset

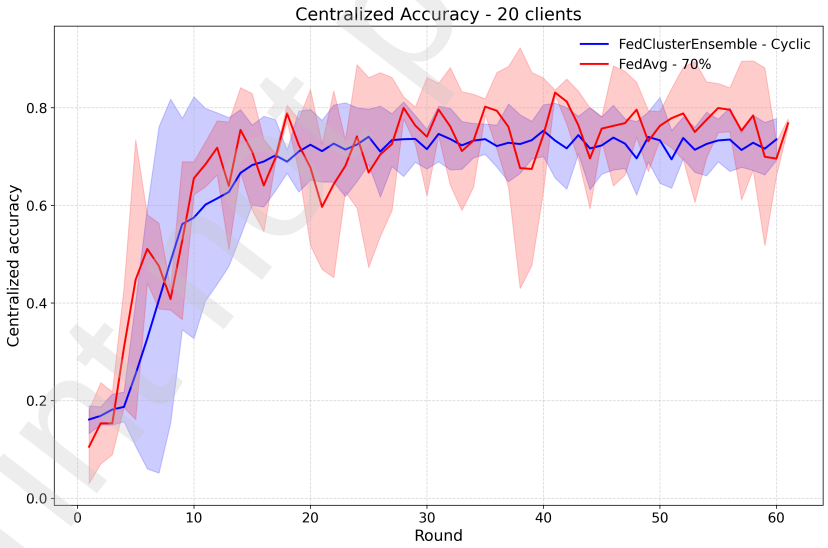

Figure 6: Centralized Accuracy Comparison of Federated Learning Strategies with 20 Clients on the SVHN Dataset

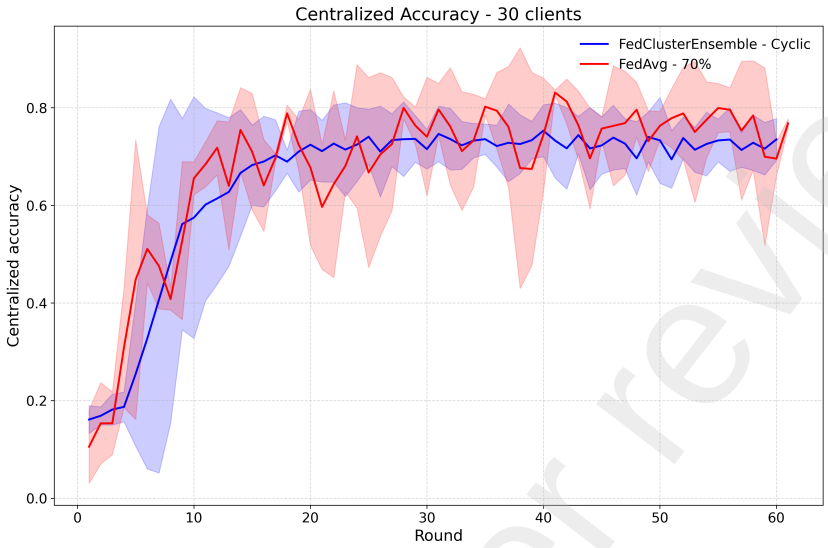

Figure 7: Centralized Accuracy Comparison of Federated Learning Strategies with 30 Clients on the SVHN Dataset

Overall, FedClusterEnsemble exhibited better stability compared to FedAvg, highlighting its suitability for handling highly non-IID data distributions. On CIFAR-10, a significant improvement in centralized accuracy was observed over the training rounds, demonstrating the effectiveness of the proposed strategy in diverse client configurations. For SVHN, while the performance of FedClusterEnsemble was slightly below that of the baseline FedAvg, it remained competitive. This difference can be attributed to specific characteristics of the dataset and underscores the importance of considering dataset properties when designing clustered federated learning approaches. Additionally, the results imply that FedClusterEnsemble may be sensitive to hyperparameter tuning, such as learning rate and batch size, which were not extensively explored in this work. Therefore, there is potential for further optimization that could improve performance on datasets like SVHN, narrowing the observed performance gap with the baseline approach.

5.3. Communication Cost Evaluation

Similarly to centralized accuracy, the communication cost of the proposed approach was evaluated against FedAvg, which was run with a fixed Fraction Fit of 70% to control the fraction of clients selected in each round.

Table 3 provides a comparison of centralized accuracy and cumulative communication costs for different strategies and datasets. Figure 8 illustrates the communication cost associated with each approach considered.

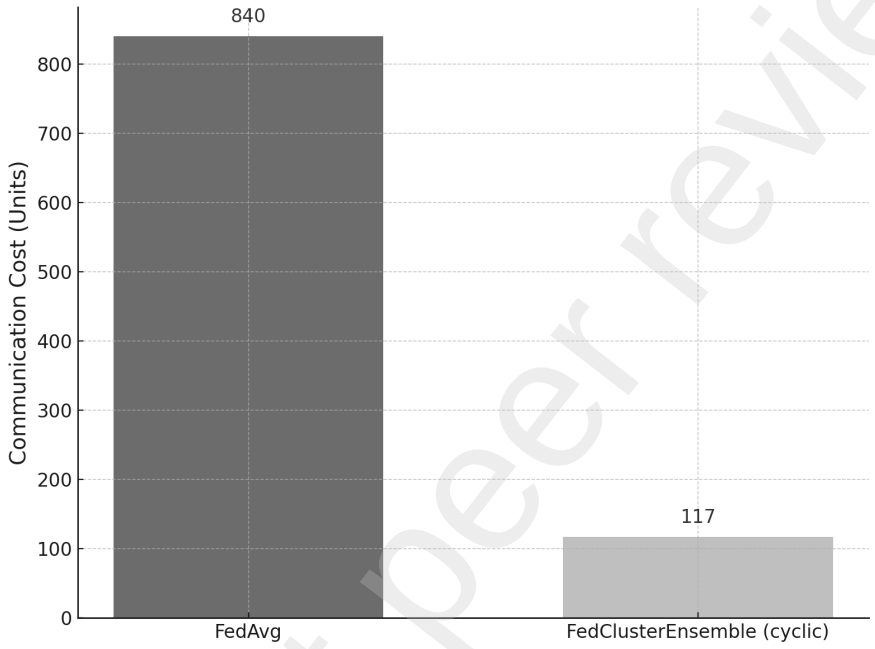

Figure 8: Communication Cost Comparison

Table 3: Comparison of Centralized Accuracy Values and Communication Costs of Federated Learning Strategies

Strategy	Dataset	30 Rounds	60 Rounds	Communication Cost
FedAvg	CIFAR-10	21%	31%	1200
FedAvg	SVHN	75%	81%	1200
FedClusterEnsemble	CIFAR-10	71%	81%	840
FedClusterEnsemble	SVHN	65%	73%	840

The **cyclic FedClusterEnsemble** strategy, which selects one client per round, demonstrates a remarkable reduction in communication cost. With only 117 units of communication cost, it achieves 45.1% centralized accuracy on CIFAR-10 and 43.2% on SVHN after 60 rounds. This significant reduction

in communication overhead is due to the strategy's ability to exploit the homogeneity within clusters and avoid redundant updates being sent to the cloud.

In contrast, FedAvg requires a communication cost of 1200 units to achieve 42.1% centralized accuracy on CIFAR-10 and 40.5% on SVHN. The proposed FedClusterEnsemble strategy, with a Fraction Fit of 70%, achieves a better balance between accuracy and communication cost, particularly in resource-constrained federated learning scenarios.

These results demonstrate that FedClusterEnsemble, particularly in its cyclic variant, offers a strong balance between communication efficiency and accuracy. By carefully selecting clients in a round-robin manner from homogeneous clusters, the cyclic approach minimizes communication overhead while preserving model quality, making it highly suitable for real-world federated learning deployments.

6. Conclusions

In this work, we introduced a novel federated learning framework, **FedClusterEnsemble**, which provides a robust and scalable solution to the challenges posed by highly non-IID data distributions. By clustering clients based on data similarity and employing a confidence-based ensemble inference mechanism, our framework not only enhances model accuracy but also significantly reduces communication overhead, making it well-suited for heterogeneous federated environments. The cyclic client selection strategy further optimizes resource usage and ensures fairness in client participation.

Our experimental results on benchmark datasets, including CIFAR-10 and SVHN, indicate that FedClusterEnsemble outperforms the baseline FedAvg method in terms of centralized accuracy and communication efficiency, particularly in scenarios with extreme data heterogeneity. Moreover, the proposed approach exhibits enhanced stability across different client configurations, which is crucial for dynamic environments.

An important aspect of our framework is its adaptability. Future work will focus on further hyperparameter optimization, dynamic clustering mechanisms, and the integration of AI agents to orchestrate adaptive client selection, monitor system performance, and enhance security. Additionally, given the inherent challenges of IoT environments, such as limited bandwidth, resource constraints, and data variability, our approach is particularly promising for real-world applications in IoT, edge computing, sensor networks, and

vehicular systems. Overall, FedClusterEnsemble lays a solid foundation for developing advanced federated learning systems capable of meeting the rigorous demands of modern, distributed, and resource-constrained applications.

7. Future Work

This paper has demonstrated the potential of FedClusterEnsemble for handling highly non-IID data in federated learning contexts, reducing communication overhead via cyclic selection, and supporting inference-only clients through a confidence-based ensemble. Nevertheless, several avenues for improvement and extension remain open:

- **Hyperparameter Optimization:** The current approach relies on DBSCAN clustering with fixed parameters (ϵ and min $\_samples$), as well as predetermined learning rate schedules and batch sizes in local training. Future work may include systematic hyperparameter tuning, possibly leveraging *Bayesian optimization* or *population-based training*, aiming to fine-tune both the clustering process and local model configurations.

- **Adaptive Clustering and Re-Clustering:** Although the proposed method performs clustering based on last-layer parameters in the initial rounds, real-world conditions often entail dynamic data distributions or the arrival of new clients over time. Investigating an approach that periodically re-clusters or merges/splits existing clusters, possibly guided by metrics of distribution drift, would enhance system adaptability.

- **Extended Data Partition Schemes:** Future evaluations could incorporate diverse partitioning strategies (e.g., Dirichlet distributions) to simulate non-IID data in more granular ways. This would clarify how FedClusterEnsemble generalizes across different types and levels of data heterogeneity.

- **Energy and Latency Considerations:** Beyond communication overhead, real-world deployments require measuring energy consumption and inference latency, especially when multiple cluster models must be queried during the inference stage. Exploring hardware-friendly pruning, quantization, or lightweight ensembles could make FedClusterEnsemble more viable on edge devices.

- **Integration of AI Agents:** A promising future direction is to incorporate intelligent agents that dynamically manage various components of the federated system:
  - *Orchestration and Hyperparameter Coaching:* AI agents could monitor training and automatically adjust hyperparameters, such as learning rates or DBSCAN parameters, based on real-time feedback (e.g., model convergence metrics or communication constraints).
  - *Adaptive Client Selection and Cluster Maintenance:* Instead of purely round-robin selection, agents might employ *reinforcement learning* to select the most beneficial clients at each round, balancing fairness, convergence speed, and communication cost. Additionally, they could automatically reassign clients to clusters or create new clusters in response to distribution shifts.
  - *Security and Anomaly Detection:* Intelligent agents could detect malicious updates or abnormal client behaviors, leveraging *anomaly detection* or *trust-based models* to isolate potential attackers or unintentional faults.
  - *Enhanced Inference-Only Support:* AI agents located on inference-only devices could provide local feedback or metrics of uncertainty, aiding the central server in reconfiguring cluster models or refining ensemble logic to better handle newly observed data.

  By introducing multi-agent intelligence, the system could become more robust, reactive, and capable of optimizing resources in scenarios where data distributions change or client behavior evolves.

- **Multi-Domain Use Cases:** While our experiments focused on CIFAR-10 and SVHN, the same mechanisms could be assessed in domains such as healthcare, autonomous vehicles, and industrial IoT. Validating the approach in such real-world contexts would reveal additional practical constraints (e.g., intermittent connectivity, memory limits) and open new possibilities for domain-specific enhancements.

Overall, the potential for combining federated learning with clustering, ensemble methods, advanced client selection, and AI-based orchestration remains vast. These future directions seek to refine the FedClusterEnsemble

framework, paving the way for an adaptive, secure, and scalable solution where heterogeneous data and resource constraints can be tackled effectively.

8. Acknowledgment

This work was conducted using the computational infrastructure of the Distributed Computing Lab at ICMC-USP (University of São Paulo). It was supported by resources funded by the São Paulo Research Foundation (FAPESP) under grants #21/06968-3, #13/07375-0, and #11/09524-7, as well as by the Center for Mathematical Sciences Applied to Industry (CeMEAI) [http://www.cemeai.icmc.usp.br/, accessed in January 2025]. Additionally, this research was funded by the National Council for Scientific and Technological Development (CNPq) under grant #405498/2021-7.

Declarations

Conflict of interest: The authors declare that they have no conflict of interest.

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

Consent of publication: Not applicable.

Competing interests: The authors declare that they have no competing interests.