Received: 13 November 2025 | Revised: 28 January 2026 | Accepted: 13 March 2026 | Published online: 13 April 2026

RESEARCH ARTICLE

Journal of Computational and Cognitive Engineering
2026, Vol. 00(00) 1–11
DOI: 10.47852/bonviewJCCE62028199

Adaptive Client-Specific Augmentation (ACSA) for Mitigating Data Heterogeneity in Federated Learning

Aiswariya Milan Kummaya[1,*], Amudha Joseph[1] and George Ghinea[2]

[1] *Department of Computer Science and Engineering, Amrita Vishwa Vidyapeetham–Bengaluru, India*

[2] *Department of Computer Science, Brunel University of London, UK*

Abstract: Federated learning (FL) enables the training of deep learning models across multiple distributed clients while maintaining data privacy. Although this approach effectively protects sensitive information, it often yields lower accuracy compared to conventional centralized training. This performance gap is primarily due to the non-independent and non-identically distributed nature of client data—a central challenge in FL known as data heterogeneity. Because each client's dataset follows a distinct distribution, local models may diverge during training, which can degrade the performance of the aggregated global model. To address this issue, various data augmentation techniques have been proposed within the FL framework. However, many of these methods still require sharing of raw data or intermediate features among clients, potentially compromising privacy and conflicting with the fundamental principles of FL. To overcome this limitation, we introduce an additional module, Adaptive Client-Specific Augmentation, within the Fed-Hetero, a refined framework that aims to identify the nature of heterogeneity present in the data and provides recommendations of augmentation policies instead of any data or feature-level information, which enhances privacy. Based on these shared policies, each client autonomously performs a dynamic data augmentation search to determine and apply the strategy most appropriate for its local dataset. This client-specific adaptation not only improves local and global model performance but also ensures strict privacy preservation. Experimental results on heterogeneous FL benchmarks demonstrate that our approach outperforms existing state-of-the-art methods in both communication efficiency and overall accuracy.

Keywords: federated learning, deep learning, data heterogeneity, data augmentation

1. Introduction

Deep learning models often require large, diverse datasets for effective training. Traditional deep learning pipelines require collecting all data into a centralized server—which is often impractical or noncompliant with data protection laws like General Data Protection Regulation (GDPR) or Health Insurance Portability and Accountability Act (HIPAA). However, in many domains such as healthcare, finance, or personalized applications, data are sensitive and cannot be centralized due to privacy, regulatory, or ethical constraints. Federated learning (FL) allows deep neural networks to be trained collaboratively across multiple clients without sharing raw data [1, 2], which helps the models to learn from a wider and more diverse set of examples and achieve better generalization and robustness across domains or user populations. However, each client's data may come from different sources, environments, or user behaviors, resulting in imbalances in label distribution, feature representation, and data quantity. This phenomenon, known as data heterogeneity, arises from variations in data distribution across participating clients. These differences can cause local models to capture client-specific patterns that do not generalize well to other clients. Consequently, when the server aggregates the local models, the global model may struggle to converge effectively, which can reduce overall performance and stability. To address the data heterogeneity challenge, various data augmentation techniques have been proposed within the FL framework. Some methods involve sharing of raw data or feature-level representations among clients [3, 4], which can improve performance but may compromise privacy. Alternatively, some domain generalization techniques [5] employ the usage of the same augmentation strategy across clients with variations in parameter values; however, this approach remains limited in addressing data heterogeneity, as it does not explore a broader range of augmentation strategies tailored to diverse client data distributions. In Fed-Hetero+ACSA, after analyzing the differences in global model performance and identifying the nature of heterogeneity present in the data, the server distributes augmentation strategies to the clients. Each client then locally selects or searches for the most appropriate transformations for its own dataset, thereby enhancing model generalization while preserving data privacy.

***Corresponding author:** Aiswariya Milan Kummaya, Department of Computer Science and Engineering, Amrita Vishwa Vidyapeetham–Bengaluru, India. Email: m_aiswariya@blr.amrita.edu

The key contributions of the study are:

1) The work investigates the effect of data heterogeneity on global model performance in non-identically distributed (non-IID) FL environments.
2) An Adaptive Client-Specific Augmentation module has been integrated into the existing Fed-Hetero [6] to communicate the augmentation policies when model performance stagnates, ensuring both computational efficiency and stable convergence.
3) Each client autonomously learns and applies a data augmentation policy suited to its local data distribution, facilitating personalized model enhancement while preserving data privacy.

2. Literature Review

Several studies have investigated methods to address data heterogeneity through improved aggregation strategies and data-level techniques. This survey presents an overview of various aggregation algorithms developed to manage different forms of heterogeneity and examines data augmentation approaches proposed to enhance model generalization and performance in heterogeneous FL environments.

The first part of the survey includes regularization- and weighted aggregation–based algorithms for handling data heterogeneity. The second part compares the performance of existing algorithms under various non-IID distributions, while the third part examines the metrics used to identify the types of heterogeneity present in the data. The final part discusses different augmentation strategies used to address data heterogeneity.

FedED [7] estimates the effective sample size required by each client to reduce the effect of label and quantity skew. The study also notes that classification accuracy declines as the number of categories increases. FedAMKD [8] uses an adaptive mutual knowledge distillation at the local level, tailored to the extent of data quantity skew on each client. This approach aims to improve both local and global model performance while reducing the negative impact of quantity-skewed heterogeneity in FL. Clients with less data contribute less to the global model, which can bias the learning process toward high-data clients. This may lead to degraded performance on low-resource or minority clients, reducing the model's generalizability and fairness across diverse data distributions. To tackle data heterogeneity in FL, FedDisk-MADE [9] uses sample-weight adjustments during local training to align local and global distributions. It preserves privacy by sharing model-based statistics instead of raw data. While privacy improves with more clients, complete privacy preservation is not guaranteed. FedDAR [10] is an FL framework designed to handle heterogeneous data across clients while improving model generalization. It combines self-supervised learning and supervised learning to learn robust feature representations. FedDAR may have performance degradation under extreme data heterogeneity or imbalance and incur a high computational cost due to simultaneous supervised and self-supervised optimization. MOON [11] compares the similarity between the client's model and the global model and improves the client's learning process through model-level contrastive learning. However, it leads to increased computation cost, higher memory use, extra communication, tuning difficulties, architecture dependence, and limited privacy protection.

FL algorithms designed for handling data heterogeneity can be classified into client and server-side algorithms [12]. Different FL algorithms for handling data heterogeneity are examined, and the results show that several approaches demonstrate strong effectiveness; their advantages often come at the expense of increased communication overhead, higher energy consumption, and knowledge sharing between clients or with the server that may compromise privacy. The impact of data skew on FL is analyzed across five aggregation algorithms—FedAvg (as the baseline), FedAvgM, FedProx, FedAdam, and FedYogi—under varying data heterogeneity scenarios [13]. While ensuring efficiency with lightweight models, the study is limited by a few communication rounds, a lack of hyperparameter tuning, and fixed Dirichlet parameters, which may affect model reliability and performance. Various challenges of heterogeneity and non-IID data in FL are analyzed in Reference [14]. It examines the performance implications while discussing practical implementation challenges and existing solutions. Various challenges in FL such as non-IID data, device-related biases, and the need to balance global accuracy with local relevance are highlighted in Reference [15]. Two key insights emerge: first, that label skew alone does not fully capture the nature of data heterogeneity; and second, that a more informative metric involves analyzing the angular distance between data subspaces across clients. Diverse and representative non-IID partitioning strategies are introduced, and an evaluation of several state-of-the-art FL algorithms is carried out in Reference [16]. Findings reveal that non-IID settings significantly impact learning performance, with no single algorithm consistently outperforming others. A major gap in current FL research is the lack of methods to identify the data distribution at client sites when there is no prior knowledge about their local datasets. The study by Jaswanth et al. [17] demonstrates that variations in client datasets significantly influence the performance of FL models. It further highlights that training models on more homogeneous data or applying suitable data augmentation strategies can enhance performance, while the choice of algorithm should be aligned with the characteristics of the underlying dataset.

The study by Jimenez-Gutierrez et al. [18] uses Hellinger distance to assess non-IID effects and evaluates five strategies across various skews. It finds that label and spatiotemporal skew significantly impact performance beyond certain thresholds. However, it lacks a detailed analysis of client-level data imbalance. Identifying and characterizing data heterogeneity in such cases remains a challenging problem, which has been addressed in Fed-Hetero [6]. Fed-Hetero examines the impact of quantity, label, and image skew on FL performance under limited data. Fed-Hetero is a self-evaluating system that identifies heterogeneity types and suggests strategies like augmentation or clustering. This enables the inclusion of clients with limited data. The work uses a uniform data augmentation applied to all clients, which does not significantly impact the global accuracy. Therefore, identifying suitable augmentation strategies tailored to each client presents an interesting direction for further research.

To effectively handle label distribution skew, SelfFed [19] was introduced, utilizing augmentative modeling with a Swin Transformer-based encoder. The relatively large model size may introduce communication delays during the fine-tuning phase. In FedRDN [20], the statistical information of the local distribution is shared between the clients. Exchanging statistical information among clients can potentially compromise data privacy. FedEqGAN [21] is a generative adversarial network (GAN) that incorporates a multi-source feature fusion mechanism. To generate high-quality synthetic data, the model integrates features from multiple data sources. The approach requires clients to access or infer information about the data distributions of other clients, which could raise privacy concerns and may not be feasible in scenarios with strict data confidentiality requirements. Fedavp [22] server distributes augmentation policies to clients, trains their models, and returns updated policies. The server aggregates these updates and redistributes the refined policies, requiring numerous communication rounds. Knowledge augmentation in FL [23] aims to enhance global model performance by allowing clients or the server to share knowledge—such as synthetic data, feature representations, or distilled information—instead of raw data, thereby handling data heterogeneity and improving generalization. However, this approach may lead to potential privacy leakage through shared representations, inconsistent knowledge quality arising from non-IID or limited client data, and increased communication and computation overhead. FedGen [24] is a personalized FL framework that leverages GANs to address data imbalance and heterogeneity by focusing on generating synthetic data rather than solely training task-specific models. However, in this setup, the simulated client datasets are uniformly sampled with equal sizes and without replacement, which diminishes the natural heterogeneity observed in real-world scenarios. As a result, the experimental setting may not accurately capture the complexity of non-IID or imbalanced data distributions, potentially limiting the generalizability of the results.

Table 1 provides an overview of the various data augmentation approaches and the metrics employed in existing studies to assess data heterogeneity.

The research gaps identified are:

1) Regularization-based methods reduce client–server divergence but often fail under strong data heterogeneity, as they increase computation while poorly aligning local and global optima.
2) Weighted aggregation methods often underweight clients with limited data, reducing their influence in the global model.
3) Existing works generally do not focus on identifying the type of heterogeneity present in client data, which limits the ability to tailor training strategies effectively.
4) Applying a fixed data augmentation strategy uniformly across all clients fails to understand client-specific data differences, often resulting in reduced improvements in both local and global model performance.
5) The existing works have a reduced focus on client-specific augmentation strategies, which could better address local data characteristics and enhance FL outcomes.
6) Server recommending data augmentation policies and clients identifying best augmentation policies based on the data statistics have not been sufficiently explored, which helps in enhancing the optimization of the global model while preserving privacy constraints.

3. Preliminaries

3.1. Federated learning

FL is a distributed learning framework in which K clients collaboratively train a shared global model without exchanging their private data. Each client $k \in \{1, 2, \ldots, K\}$ owns a local dataset

$$\mathscr{D}_k = \{(x_i^k, y_i^k)\}_{i=1}^{n_k},$$

where n_k is the number of local data samples.

The overall goal is to minimize the global loss function, which is a weighted average of all local losses:

$$\min_{w \in \mathbb{R}^d} F(w) = \sum_{k=1}^{K} p_k F_k(w), \tag{1}$$

where

$$F_k(w) = \frac{1}{n_k} \sum_{i=1}^{n_k} \ell(w; x_i^k, y_i^k), \tag{2}$$

and

$$p_k = \frac{n_k}{n}, \quad n = \sum_{k=1}^{K} n_k. \tag{3}$$

Here:

1) $w \in \mathbb{R}^d$ denotes the model parameters,
2) $\ell(w; x_i^k, y_i^k)$ is the loss function for sample (x_i^k, y_i^k),
3) $F_k(w)$ is the local objective for client k,
4) $F(w)$ is the global (federated) objective across all clients.

Thus, the global model aims to minimize the total empirical loss across all clients while keeping their data decentralized.

3.1.1. FedAvg algorithm

The **Federated Averaging (FedAvg)** [2] algorithm is the standard optimization method used in FL to solve Equation (1).

At each communication round $t = 0, 1, 2, \ldots, T-1$:

1) Server step: The central server sends the current global model w^t to a subset of clients S_t.

Table 1
Summary of federated learning approaches incorporating data augmentation techniques

Refs.	Algorithm	Data heterogeneity	Assessing heterogeneity	Augmentation
[18]	NA	Quantity skew, label skew, feature skew, spatiotemporal skew	HDD	✗
[19]	SelfFed	Label skew	✗	Swin Transformer-based encoder
[20]	FedRDN	Feature skew	✗	Input-level augmentation
[21]	FedEqGAN	NA	✗	Generates synthetic data
[22]	Fedavp	Label skew	✗	Policy sharing and aggregation
[6]	Fed-Hetero	Quantity skew, label skew, Image skew	JSD	Static augmentation

2) Client step (local training): Each selected client $k \in S_t$ initializes its local model with $w_k^{t,0} = w^t$ and performs E steps of gradient descent on its local objective:

$$w_k^{t,e+1} = w_k^{t,e} - \eta \nabla F_k(w_k^{t,e}),\ e = 0, 1, \ldots, E-1, \tag{4}$$

where η is the local learning rate.

After E local steps, client k obtains $w_k^{t,E}$ and sends it back to the server.

3) Server aggregation: The server aggregates the local models by taking a weighted average:

$$w^{t+1} = \sum_{k \in S_t} \frac{n_k}{\sum_{j \in S_t} n_j} w_k^{t,E}. \tag{5}$$

This update rule (5) represents the core of the FedAvg algorithm, where each client's contribution to the new global model is proportional to the size of its local dataset.

This iterative procedure continues for multiple communication rounds until the global model converges to a near-optimal solution of the federated objective $F(w)$.

3.2. Data heterogeneity

In FL, **data heterogeneity** (also known as **non-IID data**) refers to the situation where the local datasets across clients follow different underlying data distributions. Formally, if each client $k \in \{1, 2, \ldots, K\}$ has data sampled from its own distribution $\mathscr{P}_k$, then heterogeneity arises when

$$\mathscr{P}_1 \neq \mathscr{P}_2 \neq \ldots \neq \mathscr{P}_K.$$

This violates the standard assumption of independent and identically distributed (IID) data, where all clients would share the same distribution $\mathscr{P}$.

Data heterogeneity can lead to challenges such as slower convergence, biased global models, and increased variance in client updates.

3.2.1. Types of data heterogeneity

There are several common types of data heterogeneity in FL, as described below.

1) Feature distribution skew (covariate shift): Feature distribution skew occurs when the input features vary across different clients, but the relationship between features and labels remains the same.

   - The feature distributions differ across clients:

$$P_k(x) \neq P_{k'}(x),\ \text{but}\ P_k(y|x) = P_{k'}(y|x).$$

2) Label distribution skew (prior probability shift): Label distribution skew occurs when the proportion of labels differs across clients, even if the conditional distribution of features given labels is the same.

   - The label distributions differ across clients:

$$P_k(y) \neq P_{k'}(y),\ \text{but}\ P_k(x|y) = P_{k'}(x|y).$$

3) Quantity skew (imbalanced data sizes):

   - The number of data samples varies widely across clients:

$$n_k \ll n_{k'}\ \text{for some}\ k,\ k'.$$

4) Image skew:

   - Image data are captured with different resolutions, sensors, or preprocessing pipelines.

$$\mathscr{X}_k \neq \mathscr{X}_{k'}.$$

3.3. Dirichlet distribution

The **Dirichlet distribution** is a probability distribution that describes how a set of proportions for different categories is likely to be distributed. Each set of proportions always sums to 1, meaning it represents the relative share of each category. It models multiple categories simultaneously. A parameter controls whether the distribution is more balanced (all categories have similar proportions) or more uneven (one category dominates). It is commonly used to generate random distributions.

In FL, clients often have data that are **non-IID** (not identically distributed). The Dirichlet distribution is used to simulate the **non-IID** scenario:

1) Each client is assigned a distribution over data classes sampled from a Dirichlet distribution.
2) If the parameter is small, clients will have highly skewed data (some classes dominate, others are rare), simulating realistic heterogeneity.
3) If the parameter is large, clients will have more balanced data distributions, closer to IID conditions.
4) This helps to test and develop FL algorithms under realistic, heterogeneous data conditions.

The parameter α in the Dirichlet distribution controls the level of concentration or smoothness of the generated data proportions across different categories. The α indicates

1) When α is **small** (e.g., less than 1), the distribution becomes **sparser**. This means that one or a few categories dominate while others have very small proportions. In other words, the data are highly uneven or imbalanced.
2) When α is **large** (e.g., greater than 1), the proportions become more **balanced**. Each category tends to receive a similar share, resulting in a more uniform or homogeneous distribution.
3) When α equals 1, the distribution is approximately uniform, meaning every possible combination of proportions is equally likely.

3.4. Jensen–Shannon divergence (JSD)

The **JSD** is a symmetric and finite measure of similarity between two probability distributions. It is derived from the Kullback–Leibler (KL) divergence but improves upon it by being both symmetric and bounded. To compute it, a mean distribution is first formed by averaging the two given distributions. Then, the divergence of each distribution from this mean is calculated using the KL divergence. The JSD is obtained by taking the average of these two divergence values. It measures how much each distribution differs from its common average, providing a balanced and interpretable way to compare two probability distributions.

Further discussion on the application of JSD for identifying the type of data heterogeneity can be found in our previous work [6].

3.5. Data augmentation techniques

Data augmentation is a widely used strategy to artificially enhance the diversity of training datasets by applying transformations to existing data samples. In FL, it plays a vital role in addressing *data heterogeneity* by improving model generalization across diverse client datasets. Common augmentation operations include **jitter**, **noise addition**, **flipping**, and **cropping**. Each of these transformations introduces controlled variability into the input data, thereby enhancing robustness.

4. Methodology

In FL, multiple clients collaboratively train a global model without sharing their local data. However, the data across clients are typically non-IID, which may lead to slower convergence and stagnation in the global accuracy. To mitigate this issue, we introduce an **Adaptive Client-Specific Data augmentation (ACSA)** module to Fed-Hetero [6], which dynamically selects the optimal augmentation strategy for each client based on model performance. Figure 1 illustrates the system architecture for dynamic data augmentation designed to address data heterogeneity.

To initiate the FL process, the server distributes the initial model parameters, **w**0, to all clients.

4.1. Clients

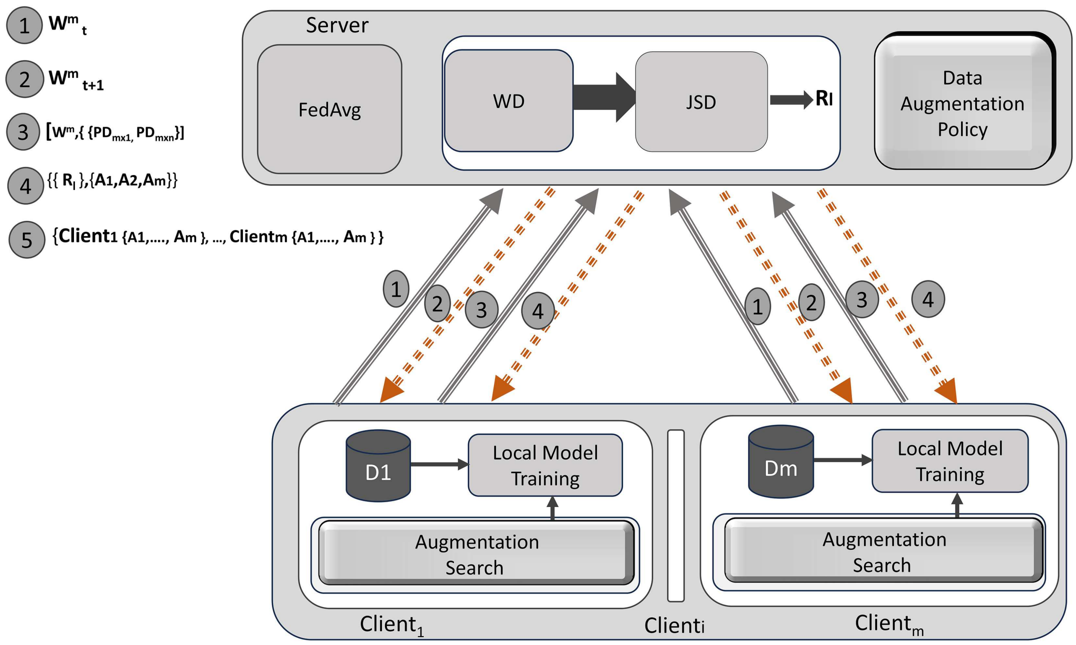
Training is performed locally by the clients, labeled from 1 to m, using their respective datasets. Each participating client returns its updated model weights to server, denoted as W_t^m (with t representing the current training round).

4.2. Server

The server aggregates these local weights using the FedAvg algorithm and broadcasts the updated global model $W^m(t+1)$ back to the clients after each round. Upon completing the specified training rounds, denoted as pt, the server calculates the weight divergence (WD) across the clients to assess the variation in model updates.

The server has a WD module, which is used to evaluate and track differences in the model updates submitted by different clients to the server. For each client in the set of clients, the algorithm computes the L2 norm (Euclidean distance) between the client's updated model weights at time $t+1$ and the corresponding global model weights. This computed divergence value represents how much the client's model differs from the global model and is added to the list of divergences. A predefined threshold θ is then set, which could either be a constant or determined dynamically. Each value that exceeds the threshold indicates whether any client's model showed significant divergence from the global model. Using WD as a basis, the server queries each client for their probabilistic data distributions, and in response, each client provides the class-wise probability distributions. Class-wise probabilistic distribution is computed by dividing the number of samples in each class by the total number of samples. PD_{mx1}, PD_{mxn} are sent to the server, where PD indicates probabilistic distributions, m denotes the client, and ×1 to ×n indicates the number of classes associated with the clients. The server then calculates the JSD between classes across clients and identifies the heterogeneity type present in their local data. In this work, a data augmentation policy module is added to the existing Fed-Hetero framework, which sends data augmentation policies to clients.

Figure 1
System architecture of Fed-Hetero+ACSA for handling data heterogeneity



4.3. Data augmentation policy

The augmentation policies are sent to the client along with the aggregated weights. The client performs an augmentation search to identify a suitable augmentation strategy.

Let $\mathscr{A} = \{a_1, a_2, \ldots, a_n\}$ be the augmentation pool containing n possible augmentation operations (e.g., rotation, flipping, color jitter, etc.). Each client i tests all augmentations in $\mathscr{A}$ on its local data D_i and evaluates their respective accuracies $acc_i(a_j)$.

The optimal augmentation strategy for client i is selected as:

$$a_i^* = \arg\max_{a_j \in \mathscr{A}} acc_i(a_j)$$

After selecting the best augmentation a_i^*, the local dataset of client i is transformed as:

$$D_{i'} = a_i^*(D_i)$$

Each client then trains the global model M_g^r locally on $D_{i'}$ to obtain updated weights w_i^{r+1}:

$$w_i^{r+1} = \text{Train}(M_g^r, D_{i'})$$

The global model for the next communication round is aggregated as:

$$M_g^{r+1} = \frac{1}{N} \sum_{i=1}^{N} w_i^{r+1}$$

Algorithm 1: Server-Side Federated Learning with Adaptive Data Augmentation (Fed-Hetero + ACSA)

Input:
N — Number of clients, R — Total global rounds, pt — Predefined round to check divergence, p — Patience threshold
Output: Final global model $\mathbf{w}$ and augmentation strategies $\{P_i\}$
Initialize:
Set global model $\mathbf{w} = \mathbf{w}_0$, best accuracy $Acc_{best} = 0$, counter $c = 0$
for each global round $t = 1$ to R
    Server distributes global model $\mathbf{w}$ to all clients
    Server receives the updated weight parameters from clients and aggregates the updates using FedAvg:

$$\mathbf{w} \leftarrow \frac{1}{N} \sum_{i=1}^{N} w_i$$

    if $t == pt$ **then**
        Server invokes Weight_Divergence(W_{clients}) to compute model divergence [6]
        if divergence is significant **then**
            Server requests for the probability distributions from client.
            Identify type of data heterogeneity using JSD:
                If label distribution varies → Label skew (R_l)
            Send heterogeneity recommendations $\{R_l\}$
            Send augmentation policy P_i to respective clients
        end if
    end if
end for
Output: Trained global model $\mathbf{w}$ and augmentation strategies $\{P_i\}$ for each client

Algorithm 2: Client-Side Local Training and Augmentation Selection

Input: N — Number of clients, α — Dirichlet parameter for data split, Global model G, local dataset D_i, augmentation strategies $\mathscr{A}$, device d.
Split the dataset into N non-IID subsets using Dirichlet(α)-> D_i
Copy model: $M_i \leftarrow G$
Each client performs local training with the Model M_i for E epochs on its local dataset and shares the updated weight to the server
Based on the server request the client sends the PD_{mx1},....,PD_{mxn} to the server
if augmentation policy received **then**
    Initialize best accuracy $Acc_{best} = 0$
    for each augmentation policy $a \in \mathscr{A}$ **do**
        Apply transformation T_a to D_i
        Evaluate M_i on D_i using T_a to get Acc_a
        if $Acc_a > Acc_{best}$ **then**
            $Acc_{best} \leftarrow Acc_a$
            Select $a_{best} \leftarrow a$
        end if
    end for
else
    Set a_{best} = None
end if
Train model M_i for E epochs using transformation $T_{a_{best}}$:

$$\min_{\theta_i} \frac{1}{|D_i|} \sum_{(x,y) \in D_i} \mathscr{L}(f_{\theta_i}(x), y)$$

Evaluate M_i after training and send model weights w_i to server
Output: Local model parameters w_i and selected augmentation a_{best}

This process repeats for each round until convergence or the specified number of rounds is reached. The server-side and client-side algorithm with adaptive data augmentation is shown in Algorithms 1 and 2.

5. Experiment Setup

To evaluate the model's robustness under non-IID data conditions, experiments were conducted by distributing CIFAR-10/100 datasets [8] among clients with varying degrees of data heterogeneity. The data allocation was done using a Dirichlet distribution with a hyperparameter of $\alpha = 0.1$, following the configuration adopted in Reference [22]. A lower α value indicates a greater level of non-uniformity across client data. For each communication round, 10 clients are randomly sampled to take part in the training process. A conventional Convolutional Neural Network (CNN) architecture is utilized for experimentation, consistent with those implemented in previous works [22]. In this study, local training at each client is conducted using a batch size of 32. The optimization is performed using the Stochastic Gradient Descent (SGD) algorithm with a learning rate of 0.01 and a momentum value of 0.9 to ensure stable and efficient convergence. Each client trains its local model for 2 epochs before transmitting the updated parameters to the central server for aggregation within the FL framework. Other datasets included in the study are Riga [25], Dhristi [26], Rim-one [27], and HRF [28]. A standard CNN architecture is employed for the experiments, in alignment with the models used in earlier research studies [6]. These datasets consist of two classes: class 0 for non-glaucoma and class 1 for glaucoma. The experiment was conducted for 100 rounds. The details of the training parameters are mentioned in Table 2.

Table 2
Federated learning configuration parameters

Description	Default value
num_clients	10
clients per round	10
alpha	0.1
number of rounds	100
client resources	{'num_cpus': 2, 'num_gpus': 1}
learning_rate	0.01
batch_size	32
optimizer	SGD
algorithm	FedAvg

6. Results and Discussions

The study focuses on determining the most suitable data augmentation strategy for each client rather than applying a uniform augmentation to all clients. Each client selects or searches for an optimal augmentation policy from a list of policies provided by the server to enhance its local performance and, consequently, the overall FL performance.

To simulate a non-IID scenario, the datasets are partitioned among clients using a Dirichlet distribution with $\alpha = 0.1$. The resulting class distributions for CIFAR-10 and CIFAR-100 are shown in Figures 2 and 3, respectively. These figures illustrate the degree of data heterogeneity across different datasets. A smaller α value, such as 0.1, generates a strongly non-IID partition, where each client's local dataset is biased toward a subset of classes. The observed variations across datasets indicate that the same partitioning strategy can produce differing levels of imbalance depending on dataset complexity and class granularity. Visualization helps us to assess the extent of non-IIDness in FL experiments and also helps in evaluating the robustness of algorithms under diverse data distributions.

To provide a more comprehensive analysis, we expand on our earlier findings in Reference [6] by including additional datasets and evaluating the robustness of JSD to understand the type of data heterogeneity present across clients. JSD quantifies the similarity (or dissimilarity) between the label distributions of different clients, providing an interpretable metric for analyzing how diverse or skewed the local datasets are relative to each other. A lower JSD value indicates that clients share more similar label distributions, reflecting a more IID-like setup, whereas higher JSD values suggest significant non-IIDness, where client data distributions differ substantially. Thus, JSD serves as an effective tool to characterize and compare the level of heterogeneity across various datasets and partitioning strategies in FL experiments.

Tables 3 and 4 show the JSD values between CIFAR-10 and CIFAR-100 samples in the client. It is observed that in the case of CIFAR-100, the client pairs (4, 5), (5, 6), and (3, 5) exhibit higher divergence, which is likely to have a greater impact on the overall accuracy of the FL process. Similarly, for CIFAR-10, the pairs (3, 7), (7, 10), and (3, 6) show significant distributional differences, suggesting that these client interactions may have contributed to performance variations in the global model.

JSD helps the client to identify the nature of data heterogeneity present in its local data.

Based on this information, the server provides tailored recommendations and augmentation policies to the clients. Each client then evaluates and selects the most suitable augmentation

Figure 2
Class distribution of CIFAR10 across clients with $\alpha = 0.1$

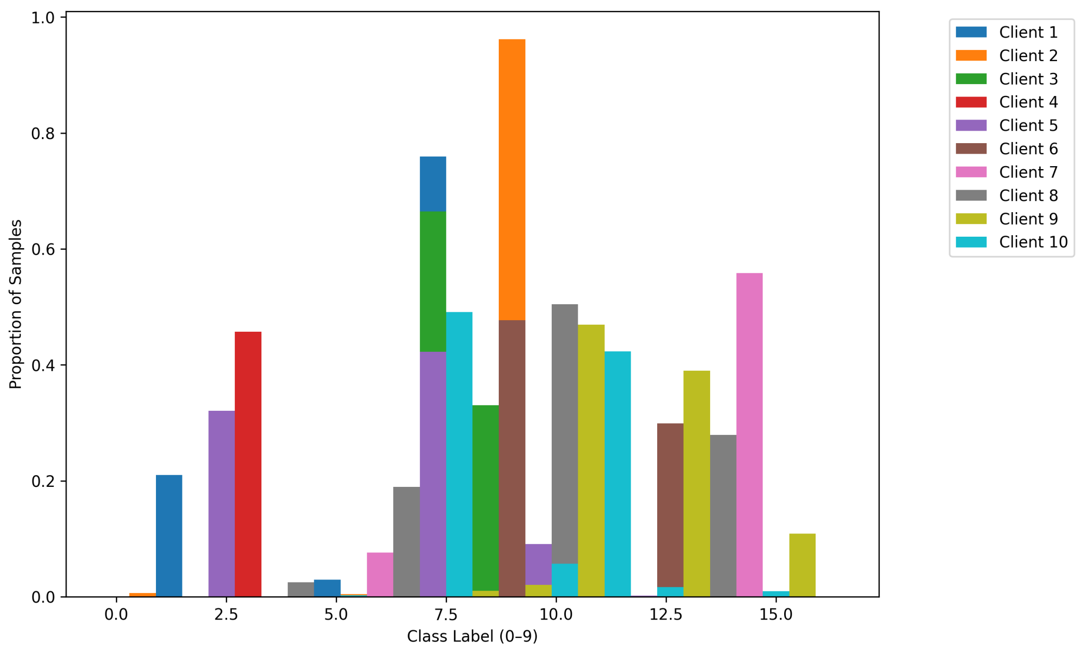


Figure 3
Class distribution of CIFAR100 across clients with $\alpha = 0.1$

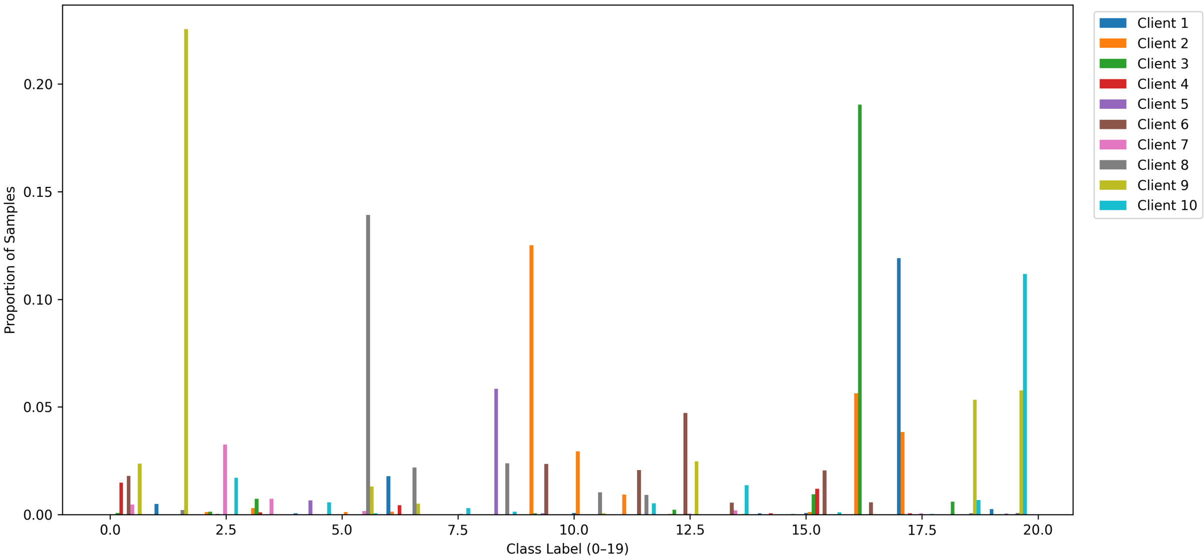


Table 3
Pairwise Jensen–Shannon divergence (JSD) between clients (CIFAR-10, $\alpha = 0.1$)

Client	Client 1	Client 2	Client 3	Client 4	Client 5	Client 6	Client 7	Client 8	Client 9	Client 10
Client 1	0.0000	0.7096	0.9497	0.6796	0.6279	0.5703	0.9631	0.5236	0.3847	0.8842
Client 2	0.7096	0.0000	0.9295	0.7412	0.8600	0.6190	0.9375	0.7171	0.5201	0.3335
Client 3	0.9497	0.9295	0.0000	0.7382	0.7850	0.9703	0.9817	0.9645	0.9588	0.5067
Client 4	0.6796	0.7412	0.7382	0.0000	0.7713	0.5490	0.8325	0.7404	0.5373	0.7149
Client 5	0.6279	0.8600	0.7850	0.7713	0.0000	0.5070	0.7641	0.8129	0.8221	0.8274
Client 6	0.5703	0.6190	0.9703	0.5490	0.5070	0.0000	0.9324	0.3244	0.4647	0.5829
Client 7	0.9631	0.9375	0.9817	0.8325	0.7641	0.9324	0.0000	0.9450	0.7998	0.9717
Client 8	0.5236	0.7171	0.9645	0.7404	0.8129	0.3244	0.9450	0.0000	0.2889	0.8356
Client 9	0.3847	0.5201	0.9588	0.5373	0.8221	0.4647	0.7998	0.2889	0.0000	0.7545
Client 10	0.8842	0.3335	0.5067	0.7149	0.8274	0.5829	0.9717	0.8356	0.7545	0.0000

Table 4
Pairwise Jensen–Shannon divergence (JSD) between clients (CIFAR-100, $\alpha = 0.1$)

Client	Client 1	Client 2	Client 3	Client 4	Client 5	Client 6	Client 7	Client 8	Client 9	Client 10
Client 1	0.0000	0.7675	0.8442	0.7908	0.7551	0.8292	0.7115	0.8214	0.7865	0.7659
Client 2	0.7675	0.0000	0.7750	0.7860	0.7502	0.8379	0.6930	0.7822	0.8102	0.7638
Client 3	0.8442	0.7750	0.0000	0.8491	0.8655	0.7929	0.7673	0.8024	0.7961	0.7985
Client 4	0.7908	0.7860	0.8491	0.0000	0.8894	0.8329	0.7832	0.8544	0.7709	0.7774
Client 5	0.7551	0.7502	0.8655	0.8894	0.0000	0.8715	0.8276	0.7600	0.7839	0.7940
Client 6	0.8292	0.8379	0.7929	0.8329	0.8715	0.0000	0.7145	0.8632	0.8114	0.7505
Client 7	0.7115	0.6930	0.7673	0.7832	0.8276	0.7145	0.0000	0.8001	0.8213	0.7926
Client 8	0.8214	0.7822	0.8024	0.8544	0.7600	0.8632	0.8001	0.0000	0.7859	0.7366
Client 9	0.7865	0.8102	0.7961	0.7709	0.7839	0.8114	0.8213	0.7859	0.0000	0.7163
Client 10	0.7659	0.7638	0.7985	0.7774	0.7940	0.7505	0.7926	0.7366	0.7163	0.0000

strategy for its specific data characteristics. Table 5 presents the improvement in accuracy observed after applying the dynamic augmentation selected by each client across different communication rounds, demonstrating that as clients adopt augmentation strategies best aligned with their data, the overall federated model performance improves progressively.

Table 5
Performance of FL for different rounds with $\alpha = 0.1$ after client-specific adaptive augmentation

Dataset	20	30	50	100
CIFAR-10	60.54	66.11	67.18	68.05
CIFAR-100	35.5	37.69	40.2	40.78
Glaucoma	73.29	71.76	72.33	76.66

Table 6
Client-specific augmentation strategies selected by each client

Dataset	Client ID	Augmentation strategy
CIFAR-10	Client 1	jitter
	Client 2	noise
	Client 3	jitter
	Client 4	jitter
	Client 5	jitter
	Client 6	jitter
	Client 7	jitter
	Client 8	jitter
	Client 9	noise
	Client 10	jitter
CIFAR-100	Client 1	crop
	Client 2	flip
	Client 3	crop
	Client 4	flip
	Client 5	jitter
	Client 6	jitter
	Client 7	crop
	Client 8	crop
	Client 9	crop
	Client 10	noise
Glaucoma	Client 1	jitter
	Client 2	flip
	Client 3	flip
	Client 4	flip

Figure 4
Local accuracy of clients before and after augmentation for CIFAR-10

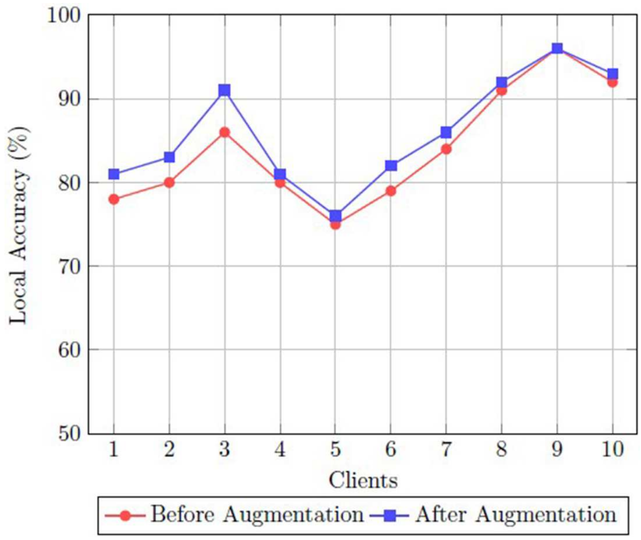


Table 6 shows the augmentation policy selected by each client after a specified number of rounds. The local accuracy of the clients before and after augmentation is shown in Figure 4. Hence, the diversity in the selected policies effectively improves the performance of the FL setup.

The proposed framework can also be evaluated on other studies, using different glaucoma datasets [29], to identify and analyze various types of data skews.

6.1. Comparison with existing works

A systematic evaluation was carried out between the Fed-Hetero+ACSA framework and existing state-of-the-art methods to evaluate its performance under heterogeneous data conditions. Our approach was compared with FedAVP [22], which requires multiple local epochs to achieve stable convergence and satisfactory accuracy. FedProx [30] emphasizes optimization stability but does not address the underlying causes of data heterogeneity. FedDyn+ [31] suffers from slow convergence and needs a large number of communication rounds. FedMix [32] requires the exchange of mixed data statistics or aggregated Mixup representations between the server and clients. Table 7 presents a comparison of Fed-Hetero+ACSA with existing approaches. The proposed Fed-Hetero+ACSA framework aims to identify the nature of data heterogeneity and suggest appropriate data augmentation strategies for the clients, enabling each client to determine dynamic, client-specific augmentations that are best suited for their local data.

Table 7
Comparison of Fed-Hetero+ACSA with existing works

Reference	Algorithm	Augmentation	Characterizing heterogeneity
[22]	FedAvp	Static	No
[30]	Fedprox	Static	No
[31]	FedDyn+	Static	No
Proposed work	Fed-Hetero+ACSA	Dynamic	Yes

7. Conclusion

In FL setup, clients collect data from diverse sources, sensors, or environments, resulting in variations in class distribution, feature representation, and sample quantity. Such discrepancies lead to model divergence, slower convergence, and degraded global performance, as each client's model tends to overfit its local data characteristics. To address this issue, data augmentation has emerged as an effective strategy. Synthetically expanding local datasets through data augmentation improves data diversity and reduces the statistical gap among clients. An adaptive client-specific augmentation policy allows each client to generate augmented samples suited to its data distribution, thereby improving local generalization and contributing to a more robust global model. The results show that integrating data augmentation into FL training helps to reduce the adverse effects of heterogeneity and enhances both fairness and global accuracy across clients.

Acknowledgment

The authors sincerely acknowledge the support provided by the "Artificial Intelligence for Health and Well-being" Thrust Area Group within the Department of Computer Science and Engineering, Amrita School of Computing, and the Eye Tracking and Computer Vision Research Lab for their invaluable assistance in carrying out this research. The authors also gratefully acknowledge Sri Mata Amritanandamayi Devi (Amma), Chancellor, Amrita Vishwa Vidyapeetham, for her inspiration and for providing financial support for the Article Processing Charges (APC) of this publication.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

The CIFAR-10 and CIFAR-100 data that support the findings of this study are openly available at https://www.cs.toronto.edu/~kriz/cifar.html; The Riga data that support the findings of this study are openly available at https://deepblue.lib.umich.edu/data/concern/data_sets/3b591905z; The Dhristi data that support the findings of this study are openly available at https://www.kaggle.com/datasets/lokeshsaipureddi/drishtigs-retina-dataset-for-onh-segmentation; The Rim-one data that support the findings of this study are openly available at https://github.com/miag-ull/rim-one-dl; and The HRF data that support the findings of this study are openly available at https://www5.cs.fau.de/research/data/fundus-images.

Author Contribution Statement

Aiswariya Milan Kummaya: Conceptualization, Methodology, Software, Investigation, Resources, Data curation, Writing - original draft, Visualization. **Amudha Joseph:** Conceptualization, Methodology, Validation, Formal analysis, Writing – review & editing, Supervision, Project administration. **George Ghinea:** Supervision, Project administration.

References

[1] Kairouz, P., & McMahan, H. B. (2021). Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, *14*(1–2), 1–210. https://doi.org/10.1561/2200000083

[2] Yurdem, B., Kuzlu, M., Gullu, M. K., Catak, F. O., & Tabassum, M. (2024). Federated learning: Overview, strategies, applications, tools and future directions. *Heliyon*, *10*(*19*), e38137. https://doi.org/10.1016/j.heliyon.2024.e38137

[3] Zhou, T., Yuan, Y., Wang, B., & Konukoglu, E. (2024). Federated feature augmentation and alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *46*(*12*), 11119–11135. https://doi.org/10.1109/tpami.2024.3457751

[4] Zhu, H., Fan, Y., & Xie, Z. (2025). UA-PDFL: A personalized approach for decentralized federated learning. *Neurocomputing*, *657*, 131579. https://doi.org/10.1016/j.neucom.2025.131579

[5] de Luca, A. B., Zhang, G., Chen, X., & Yu, Y. (2022). *Mitigating data heterogeneity in federated learning with data augmentation*. arXiv. https://doi.org/10.48550/arXiv.2206.09979

[6] Milan Kummaya, A., Joseph, A., Rajamani, K., & Ghinea, G. (2025). Fed-Hetero: A self-evaluating federated learning framework for data heterogeneity. *Applied System Innovation*, *8*(2),28. https://doi.org/10.3390/asi8020028

[7] Zhou, Y., Wang, J., Qin, Y., Kong, X., Xie, X., Qi, H., & Zeng, D. (2025). Federated Learning with complete service commitment of data heterogeneity. *Knowledge-Based Systems*, *310*, 112937. https://doi.org/10.1016/j.knosys.2024.112937

[8] Ge, S., Liu, D., Yang, Y., He, J., Zhang, S., & Cao, Y. (2024). FedAMKD: Adaptive mutual knowledge distillation federated learning approach for data quantity-skewed heterogeneity. In *2024 IEEE International Conference on Systems, Man, and Cybernetics,* 4710–4715. https://doi.org/10.1109/SMC54092.2024.10831995

[9] Nguyen, H., Wu, P., & Chang, J. M. (2024). Federated learning for distribution skewed data using sample weights. *IEEE Transactions on Artificial Intelligence*, *5*(*6*), 2615–2626. https://doi.org/10.1109/TAI.2023.3348073

[10] Kwak, Y., & Jung, M. (2025). FedDAR: Federated learning with data-quantity aware regularization for heterogeneous distributed data. *IEEE Access*, *13*, 133208–133217. https://doi.org/10.1109/access.2025.3591839

[11] Li, Q., He, B., & Song, D. (2021). Model-contrastive federated learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10708–10717. https://doi.org/10.1109/cvpr46437.2021.01057

[12] Mora, A., Bujari, A., & Bellavista, P. (2024). Enhancing generalization in Federated Learning with heterogeneous data: A comparative literature review. *Future Generation Computer Systems*, *157*, 1–15. https://doi.org/10.1016/j.future.2024.03.027

[13] Nascimento, L., Awaysheh, F. M., & Alawadi, S. (2024). Data skew in federated learning: An experimental evaluation on aggregation algorithms. In *2024 2nd International Conference on Federated Learning Technologies and Applications*, 162–170. https://doi.org/10.1109/FLTA63145.2024.10840118

[14] Karami, M., & Karami, A. (2025). Harmony in federated learning: A comprehensive review of techniques to tackle heterogeneity and non-IID data. *Cluster Computing*, *28*(*9*), 570. https://doi.org/10.1007/s10586-025-05250-y

[15] Vahidian, S., Morafah, M., Chen, C., Shah, M., & Lin, B. (2024). Rethinking data heterogeneity in federated learning: Introducing a new notion and standard benchmarks. *IEEE Transactions on Artificial Intelligence*, *5*(*3*), 1386–1397. https://doi.org/10.1109/tai.2023.3293068

[16] Li, Q., Diao, Y., Chen, Q., & He, B. (2022). Federated learning on non-IID data silos: An experimental study. In *2022 IEEE 38th International Conference on Data Engineering*, 965–978. https://doi.org/10.1109/icde53745.2022.00077

[17] Jaswanth, M., Narayana, N. K. L., Rahul, S., Kummaya, A. M., & Joseph, A. (2023). Emotion and advertising effectiveness: A novel facial expression analysis approach using federated learning. In *2023 IEEE 20th India Council International Conference*, 368–373. https://doi.org/10.1109/indicon59947.2023.10440766

[18] Jimenez-Gutierrez, D. M., Hassanzadeh, M., Anagnostopoulos, A., Chatzigiannakis, I., & Vitaletti, A. (2026). A thorough assessment of the non-IID data impact in federated learning. *Journal of Industrial Information Integration*, *50*, 101052. https://doi.org/10.1016/j.jii.2025.101052

[19] Ali Khowaja, S., Dev, K., Muhammad Anwar, S., & George Linguraru, M. (2025). SelfFed: Self-supervised federated learning for data heterogeneity and label scarcity in medical images. *Expert Systems with Applications*, *261*, 125493. https://doi.org/10.1016/j.eswa.2024.125493

[20] Yan, Y., Fu, H., Li, Y., Xie, J., Ma, J., Yang, G., & Zhu, L. (2025). A simple data augmentation for feature distribution skewed federated learning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 25749–25758. https://doi.org/10.1109/CVPR52734.2025.02398

[21] Xiao, Y., Zhao, D., Li, X., Li, T., Wang, R., & Wang, G. (2025). A federated learning-based data augmentation method for privacy preservation under heterogeneous data. *IEEE Transactions on Mobile Computing*, *24*(*9*), 8105–8118. https://doi.org/10.1109/tmc.2025.3553501

[22] Hong, M., Yun, J., Jeon, I., & Kim, G. (2024). FedAvP: Augment local data via shared policy in federated learning. In *38th Conference on Neural Information Processing Systems*, 18090–18121. https://doi.org/10.52202/079017-0575

[23] Wu, W., He, L., Long, S., Abdelmoniem, A. M., Wu, Y., Mao, R., & Li, K. (2025). *Task-agnostic federation over decentralized data: Research landscape and visions*. arXiv Preprint.

[24] Zhao, P., Guo, S., Li, Y., Yang, S., & Ren, X. (2025). FedGen: Personalized federated learning with data generation for enhanced model customization and class imbalance. *Future Generation Computer Systems*, *164*, 107595. https://doi.org/10.1016/j.future.2024.107595

[25] Almazroa, A., Alodhayb, S., Osman, E., Ramadan, E., Hummadi, M., Dlaim, M., ..., & Lakshminarayanan, V. (2018). Retinal fundus images for glaucoma analysis: The RIGA dataset. In *Medical Imaging 2018: Imaging Informatics for Healthcare, Research, and Applications*, 105790B. https://doi.org/10.1117/12.2293584

[26] Sivaswamy, J., Krishnadas, S. R., Datt Joshi, G., Jain, M., & Syed Tabish, A. U. (2014). Drishti-GS: Retinal image dataset for optic nerve head(ONH) segmentation. In *2014 IEEE 11th International Symposium on Biomedical Imaging*, 53–56. https://doi.org/10.1109/isbi.2014.6867807

[27] Fumero, F., Alayon, S., Sanchez, J. L., Sigut, J., & Gonzalez-Hernandez, M. (2011). RIM-ONE: An open retinal image database for optic nerve evaluation. In *2011 24th International Symposium on Computer-Based Medical Systems*, 1–6. https://doi.org/10.1109/cbms.2011.5999143

[28] Budai, A., Bock, R., Maier, A., Hornegger, J., & Michelson, G. (2013). Robust vessel segmentation in fundus images. *International Journal of Biomedical Imaging*, *2013*(1), 154860. https://doi.org/10.1155/2013/154860

[29] Yoshitha, C., Prakashini, G. V., Neha, S., & Kummaya, A. M. (2025). Glaucoma insights: A visual comparative analysis of fundus images across diverse datasets. In *2025 3rd International Conference on Smart Systems for Applications in Electrical Sciences*, 1–7. https://doi.org/10.1109/icsses64899.2025.11009482

[30] Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., & Smith, V. (2019). Federated optimization for heterogeneous networks. In *Proceedings of the 1st Adaptive & Multitask Learning Workshop*.

[31] Mai, V. S., La, R. J., & Zhang, T. (2024). A study of enhancing federated learning on non-IID data with server learning. *IEEE Transactions on Artificial Intelligence, 5*(11), 5589–5604. https://doi.org/10.1109/tai.2024.3430250

[32] Yoon, T., Shin, S., Hwang, S. J., & Yang, E. (2021). FedMix: Approximation of mixup under mean augmented federated learning. In *9th International Conference on Learning Representations*.

How to Cite: Kummaya, A. M., Joseph, A., & Ghinea, G. (2026). Adaptive Client-Specific Augmentation (ACSA) for Mitigating Data Heterogeneity in Federated Learning. *Journal of Computational and Cognitive Engineering*. https://doi.org/10.47852/bonviewJCCE62028199