Overcoming Data Imbalance in Federated Learning with Calibration Weighting

Anonymous*, Anonymous*, and Anonymous*, *Member, IEEE*
*Anonymous

Abstract—**Federated Learning (FL) is a crucial technique in data mining, enabling machine learning across decentralized devices while preserving data privacy. A major challenge in FL is data imbalance, where underrepresented classes lead to biased models and poor performance. This paper introduces a novel FL approach incorporating calibration weighting to enhance model performance and data integrity. Our method addresses data imbalance through calibration resampling techniques and calibrated loss functions, aligning model training with true data distributions. Extensive experiments on datasets like MNIST, CIFAR10, and Adult Income demonstrate significant improvements in accuracy and loss. These findings provide the potential of calibration weighting in FL, offering a robust solution for effective distributed machine learning.**

Index Terms—**Calibration, Differential Privacy, Federated Learning, Data Mining, Statistical Heterogeneity**

I. Introduction

FEDERATED learning (FL) represents a significant evolution in machine learning (ML), addressing contemporary challenges in data management, privacy, and scalability [1], [2], [3]. Originating from the need to process large datasets efficiently while maintaining data privacy, FL has become crucial with the proliferation of IoT devices, social media, and e-commerce, which generate vast amounts of data unsuitable for centralized ML models. This decentralized approach enhances privacy by localizing data, reducing communication overheads, and improving the scalability of ML systems.

The importance of FL lies in its innovative approach to model training. Unlike traditional ML approaches that require data centralization [4], [5], FL enables machine learning models to be trained across multiple decentralized devices (such as smartphones, wearable devices, and IoT gadgets) while keeping the data localized. This method not only addresses privacy concerns by minimizing data exposure but also tackles logistical challenges associated with managing extensive datasets.

As FL progresses, it faces the issue of uneven data distribution across various devices, known as data imbalance [6]. This imbalance, reflecting real-world variability, can significantly affect the accuracy and impartiality of FL-trained models. Addressing this issue is crucial to fully leverage FL's capabilities, as it influences the effectiveness and fairness of the models [3], [7], [8], [9]. Particularly, it impacts classification performance, leading to biased models that perform well on majority classes but poorly on minority classes [10]. Thus, creating solutions to balance data in FL systems is a key research priority.

This paper introduces a novel methodology that addresses data imbalance while enhancing the security and privacy aspects of FL. By leveraging calibration weighting and resampling techniques, we aim to create a more robust and equitable FL framework, contributing to the broader field of data mining and privacy-preserving ML. Our work builds on recent advances in privacy-preserving FL, such as differential privacy and homomorphic encryption [11], [12], [13], [14], [15], highlighting the need for secure and private FL methodologies.

II. Related Work

A. FL with Skewed Data Distribution

The pioneering work in Federated Learning (FL) is Federated Averaging (FedAvg) proposed by McMahan et al. [1]. FedAvg has become a standard in the FL field due to its simplicity and effectiveness. The core idea of FedAvg is to train local models on clients' devices using their own data and then send these local models to a central server where they are averaged to update the global model. This process is iteratively repeated, with the global model being distributed back to clients for further local training. Despite its initial success, FedAvg struggles with skewed data distributions [9], a common scenario in real-world applications where data is not identically distributed across clients. Addressing this challenge is crucial for data mining applications that require robust and unbiased models across diverse datasets.

On the client side, methods such as FedProx [7] and SCAFFOLD [16] have been proposed to mitigate the skewed data issue. FedProx introduces a proximal term to the local training objective, constraining local models to be closer to the global model, thereby limiting the impact of data heterogeneity. SCAFFOLD corrects client updates using control variates, which helps reduce variance caused by data discrepancies among clients. Both methods aim to align local model updates more closely with the global model, enhancing convergence under imbalanced conditions. These approaches are essential in data mining to ensure that local patterns and trends are accurately captured and aggregated into the global model.

Server-side aggregation has also seen significant advancements. FedNova [17] and FedMA [18] are notable methods addressing client update aggregation. FedNova normalizes client updates to ensure equitable contribution from each client, managing diversity in data and computational resources. FedMA implements layer-wise aggregation of neural network parameters, aligning and averaging corresponding layers from

Corresponding author: Anonymous

different clients to construct a more representative global model. However, these methods introduce challenges such as increased computational complexity and communication overhead. Addressing these limitations is crucial for scalable data mining in federated settings.

Our study contributes to this area by proposing a novel approach tailored to highly-skewed data distributions. We focus on enhancing server-side aggregation to achieve a more balanced and representative global model while minimizing complexity and communication overhead. This addresses current gaps in FL research, ensuring that data mining techniques can be effectively applied in federated environments.

B. Privacy and Data Security in FL

Federated Learning has seen significant advances in privacy and data security, critical for data mining applications involving sensitive information. Differential Privacy (DP) adds noise to data or gradients to protect individual privacy while allowing accurate aggregate analysis [11], [15]. DP ensures that the contribution of any single data point remains indistinguishable, thus protecting user privacy.

Homomorphic Encryption (HE) enables computations on encrypted data without needing decryption, ensuring data remains confidential throughout the learning process. HE effectively prevents privacy leakage from gradients, addressing challenges such as poisoning attacks [13].

Secure Multi-Party Computation (SMC) allows multiple parties to jointly compute functions over their inputs while keeping those inputs private. Decentralized FL frameworks using SMC ensure privacy and verifiability in the learning process [14]. These techniques are foundational in privacy-preserving data mining, enabling collaborative analysis without exposing raw data.

Recent studies have explored robust aggregation rules and anomaly detection mechanisms to enhance security in FL. ShieldFL integrates model poisoning defenses with privacy-preserving techniques to protect against malicious attacks while maintaining model integrity [12]. These advancements are vital for secure data mining in distributed environments.

Our proposed calibration weighting and resampling techniques further enhance data privacy in FL. By calibrating weights and resampling data, our method ensures that the model training process is less influenced by outliers and malicious data points, which can lead to privacy breaches.

C. Calibration Weighting in FL

Calibration weighting [19] is a sophisticated statistical technique used to construct adjusting weights in scenarios where certain groups are underrepresented or over-represented. Originating from survey methodology [20], [21], calibration weighting corrects biases in sample data, ensuring each data point contributes appropriately to the overall analysis. This technique is vital in data mining for addressing sample bias, improving data representativeness, and enhancing result reliability. Recent advancements have broadened its application beyond traditional survey methodologies, notably in addressing nonresponse bias within health surveys and refining pollutant concentration estimates in environmental studies [22], [23], [24]. These developments demonstrate the technique's growing relevance across diverse fields, including data mining.

Recent research has applied calibration weighting to federated learning. Zhang et al. [25] established a calibrated loss function to minimize bias in client updates, mainly for binary classification. Luo et al. [26] addressed data imbalance by creating virtual data representation following calibration statistics but retained the conventional FedAvg loss function, failing to address computation overhead. Shang et al. [27]'s FEDIC introduces server-side calibration and distillation to mitigate skewed and long-tailed data distributions, albeit with challenges such as reliance on auxiliary balanced datasets and increased computational complexity. Chen et al. [28] present CalFAT, focusing on Federated Adversarial Training to enhance stability and robustness through calibration, particularly under label skewness. Despite its effectiveness across various datasets, CalFAT's specialization in adversarial scenarios may limit its general applicability in FL.

Our approach, Federated Learning with Feature Calibration and client Re-sampling (FL-FCR), tackles skewed data distribution across clients using a two-step process. Initially, we integrate a calibrated loss function during client updates to reduce bias. Next, we compute both local and global calibration statistics to guide the server in resampling the data before the next training phase. This ensures that the model trains on data that more accurately represents the overall distribution, thereby reducing model bias. Additionally, we perform a thorough security analysis to ensure that our approach not only improves performance but also enhances the privacy and robustness of the federated learning system, crucial for secure and effective data mining.

III. Motivation and Contribution

The main contributions of this paper are summarized as follows:

- We addressed the challenge of highly-skewed data distribution, a common issue in data mining, by employing a resampling technique guided by calibration statistics for each class. This was followed by server-side retraining and the introduction of a calibrated loss function to minimize loss.
- Our analysis revealed the inadequacy of the prevalent global objective function in handling significant data imbalance. We propose a revised approach to better manage such conditions.
- Extensive experiments on real-world datasets demonstrated superior performance of our approach compared to other leading FL models under highly imbalanced data. This highlights its effectiveness in practical data mining scenarios.
- By integrating calibration weighting and resampling techniques, our approach improves model accuracy and strengthens privacy protections, essential for secure data mining in federated environments.

IV. Proposed Method

Our proposed FL-FCR consists of two phases. Initially, we enhance the local update process at the client level with an improved loss function (Section IV-C). Subsequently, the central server performs dataset resampling based on calibration statistics prior to further training (Section IV-B). The workflow of FL-FCR is illustrated in Figure 1, showing the interactions between the server and clients during training, including local training, local calibration computation, and server-side resampling and model updating. This approach addresses data imbalance effectively while enhancing model performance and maintaining privacy.

The following pseudocode outlines the steps involved in the Federated Learning with Feature Calibration and Client Resampling (FL-FCR) algorithm:

Algorithm 1 Federated Learning with Feature Calibration and Client Re-sampling (FL-FCR)

Input: T is the maximum number of training rounds, m is the number of clients selected in each training round, N_{epoch} is the number of local epochs, η is the local learning rate, v is the extracted feature from $f_\theta(x)$, and B_k is the local batch size of the k-th client

Output: Global model parameter θ_G to minimize the empirical risk

```
[Server-side]:
Initialize $\theta_G^0$    ▷ Initialize the global model parameter
Select a subset $S_t$ of $m$ clients at random; broadcast global
  parameter $\theta_G^{(t)}$ to selected clients
for each round $t$ from 1 to $T$ do
    for each client $k \in S_t$ in parallel do
        $\theta_k^{(t)} \leftarrow$ LocalTraining$(k, \theta_G^{(t)})$
    end for
end for
[Client-side]:
(1) LocalTraining$(k, \theta)$:
Divide local dataset $D_k$ into several batches; $B_k$ is the set
  of the batches in the $k$-th client
for each epoch from 1 to $N_{epoch}$ do
    for each batch $b \in B_k$ do
        $\theta \leftarrow \theta - \eta \nabla L^{CAL}(\theta; b)$    ▷ Update model using
  calibrated loss
    end for
end for
(2) Local calibration computation $(\mu, \text{cov})$:
for each client $k \in [K]$ do
    for each class $c \in [C]$ do
        Compute extracted feature $v_{c,k,i} = f_\theta(x_{c,k,i})$
        Compute calibrated mean (Eq. 3)
        Compute calibrated covariance (Eq. 4)
    end for
end for
return $\theta_k^{(t+1)}$, $\mu_{c,k}$, and $\text{cov}_{c,k}$ to the server
[Server Retraining]:
(1) Global calibration computation $(\mu, \text{cov})$:
for each class $c \in [C]$ do
    Compute calibrated mean (Eq. 5)
    Compute calibrated covariance (Eq. 6)
    Resample $D_c$ from $N(\mu_c, \text{cov}_c)$
end for
(2) Obtain new global model
Receive re-sampled dataset $D_c$ and local parameters
  $\theta_k^{(t+1)}$ from each client
$\theta_G^{(t+1)} = \sum_{k=1}^{m} \theta_k^{(t+1)}$    ▷ Aggregate local models
$\theta_G =$ retrain$(\theta_G^{(t+1)}, D_c)$    ▷ Re-train global model using
  the aggregated parameters and the re-sampled data
```

A. Standard FL Setting

The FL problem can be considered as a standard supervised ML problem, mapping input values $x_{k,j}$ to output label $y_{k,j}$ for prediction. Here, $x_{k,j}$ and $y_{k,j}$ are the $j(= 1, \ldots, n_k)$-th input feature values and outcome values collected at the $k(= 1, \ldots, K)$-th client. The input-output pairs $(x_{k,j}, y_{k,j})$ are the client k's samples stored locally, with only intermediate updates sent to the central server periodically. Assume m clients are selected in each training round; our goal is to minimize the empirical loss on input-output pairs with model parameters θ. The global objective function $F(\theta)$ is represented as:

$$\min_{\theta \in R^d} F(\theta), \quad \text{where } F(\theta) := \sum_{k=1}^{m} p_k f_k(\theta), \tag{1}$$

where $p_k = n_k/n$ denotes a fraction of data samples collected at the k-th client, $n = \sum_{k=1}^{m} n_k$. The local objective function for the k-th client over its data samples n_k is:

$$f_k(\theta) = \frac{1}{n_k} \sum_{j=1}^{n_k} L(\theta; x_{k,j}, y_{k,j}), \tag{2}$$

where $L(\theta)$ is a predetermined loss function evaluated at each sample. The global objective function Eq. (1) can be interpreted as a weighted sum of the local loss functions. Clients periodically send model updates through an aggregating server to find the parameter θ that minimizes the empirical loss.

B. Calibrated Statistics

We focus on multi-classification tasks with each client $k \in [K]$ in class $c \in [C]$. Client k has a local dataset D^k and the entire dataset is $D = \cup_{k \in [K]} D^k$. Let $D_c^k = \{(x, y) \in D^k : y = c\}$ be the set of samples with ground-truth label c in client k. Given a sample pair (x, y), a function $f_\theta(x)$ parameterized by θ maps input x into a feature vector $v = f_\theta(x) \in R^d$. A linear classifier in the last layer $s_w(x) = \{w_c v\}_{c \in y}$ parameterized by w produces probability distribution $p = \sigma(s(x))$ after the softmax function $\sigma(\cdot)$. Here, $p = \sigma(s(x))$ is the final prediction for input x.

Let $N_{c,k} = |D_c^k|$ be the number of samples of class c in client k, and $N_c = \sum_{k=1}^{K} N_{c,k}$ be the total samples of class

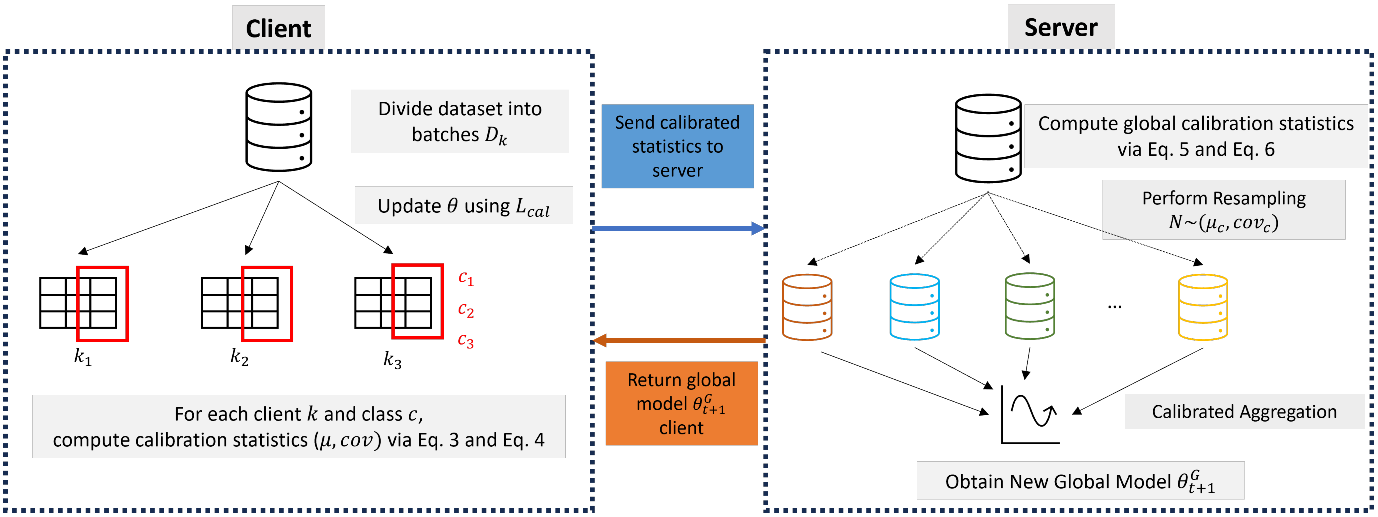

Client
Divide dataset into batches D_k
Update θ using L_{cal}
c_1
c_2
c_3
k_1
k_2
k_3
For each client k and class c, compute calibration statistics (μ, cov) via Eq. 3 and Eq. 4
Send calibrated statistics to server
Return global model θ^G_{t+1} client
Server
Compute global calibration statistics via Eq. 5 and Eq. 6
Perform Resampling $N\sim(\mu_c, cov_c)$
...
Calibrated Aggregation
Obtain New Global Model θ^G_{t+1}

Fig. 1. Illustration of our proposed method

c. During local training, client k generates a feature vector $\{v_{c,k,1}, v_{c,k,2}, \dots, v_{c,k,N_{c,k}}\}$ for class c. Local sample mean $\mu_{c,k}$ and sample covariance $\text{cov}_{c,k}$ are computed as follows:

$$\mu_{c,k} = \frac{1}{N_{c,k}} \sum_{i=1}^{N_{c,k}} v_{c,k,i}, \tag{3}$$

$$\text{cov}_{c,k} = \frac{1}{N_{c,k}-1} \sum_{i=1}^{N_{c,k}} (v_{c,k,i} - \mu_{c,k})(v_{c,k,i} - \mu_{c,k})^\top \tag{4}$$

for $N_{c,k} \geq 2$.

1) Calibrated Global Mean and Covariance: After local training, client k sends local updates $(\mu_{c,k}, \text{cov}_{c,k})$ to the server, maintaining data privacy. The server aggregates the updates and calculates global mean and covariance:

$$\mu_c = \frac{1}{N_c} \sum_{k=1}^{K} \sum_{i=1}^{N_{c,k}} v_{c,k,i} = \sum_{k=1}^{K} \frac{N_{c,k}}{N_c} \mu_{c,k}, \tag{5}$$

$$\begin{aligned} \text{cov}_c = &\sum_{k=1}^{K} \frac{N_{c,k}-1}{N_c-1} \text{cov}_{c,k} \\ &+ \sum_{k=1}^{K} \frac{N_{c,k}}{N_c-1} \mu_{c,k}\mu_{c,k}^\top \\ &- \frac{N_c}{N_c-1} \mu_c \mu_c^\top . \end{aligned} \tag{6}$$

We exclude sample covariance estimates of clients with $N_{c,k} < 2$. See Appendix for technical details. The server then resamples based on these statistics and performs retraining to enhance model performance, improving data security by mitigating the influence of outliers and malicious data points.

C. Calibrated Loss Function

Assume the classification model produces predicted label:

$$\hat{y} = \arg\max\ s(x).$$

For balanced datasets, the goal is to learn a score function $s(\cdot)$ that minimizes the misclassification error $L(y \neq \hat{y})$. Softmax cross-entropy loss is used in multi-class classification tasks:

$$\begin{aligned} L(y, \hat{y}) &= L(y, s(x)) \\ &= \log \left[\sum_{y' \in [C]} e^{s_{y'}(x)} \right] - s_y(x) \\ &= \log \left[1 + \sum_{y' \neq y} e^{s_{y'}(x) - s_y(x)} \right] . \end{aligned} \tag{7}$$

However, this loss function is unsuitable for highly imbalanced datasets [29]. Motivated by [29] and [25], we adopt the balanced error rate (BER) by averaging per-class error rates and minimizing the calibrated error:

$$e^{Cal} = \min \frac{1}{C} \sum_{y \in [C]} P(y \neq \hat{y}), \tag{8}$$

where c is the class. The calibrated error implies that the probability function $P(y|x) \propto \frac{1}{c} \cdot P(x|y)$, in contrast to $P(y|x) \propto P(y) \cdot P(x|y)$ in Eq. (7). This indicates that varying $P(y)$ does not impact the optimal results in Eq. (8) [25].

To obtain the test error bound for calibrated error in Eq. (8), consider a binary classification framework, where [30] demonstrate that the test error can be bounded by:

$$\frac{1}{\gamma_p \sqrt{N_p}} + \frac{1}{\gamma_q \sqrt{N_q}}, \tag{9}$$

where N_p and N_q are the sample sizes of classes p and q, and γ_p and γ_q are the margins of classes p and q. To make γ_p and γ_q optimal, they must satisfy:

$$\frac{1}{\gamma_p \sqrt{N_p}} + \frac{1}{\gamma_q \sqrt{N_q}} \geq \frac{1}{(\gamma_p - \delta) \sqrt{N_p}} + \frac{1}{(\gamma_q + \delta) \sqrt{N_q}}, \tag{10}$$

where $\gamma_p, \gamma_q > 0$ and $\delta \in (-\gamma_q, \gamma_p)$. Assuming classifiers $s \in S$ can achieve a total sum of margins $\gamma_p + \gamma_q = \beta$, after applying margin-based generalization bound (Theorem 2 in [31]), there exists a classifier $s^\star(\cdot)$ in the last layer:

$$\gamma_p^\star = \frac{\beta N_q^{1/4}}{N_p^{1/4} + N_q^{1/4}}, \quad \gamma_q^\star = \frac{\beta N_p^{1/4}}{N_p^{1/4} + N_q^{1/4}}. \tag{11}$$

Next, we extend to the multi-class setting of the hinge loss, where the optimal label margin in class c is:

$$\Delta_c = \frac{H}{N_c^{1/4}}, \text{ for } c \in \{1, \cdots, C\}, \tag{12}$$

where H is a hyper-parameter to be tuned. As discussed in Section IV-C, the calibrated loss can be bounded by Eq. (9).

Inspired by [32], we apply L_2 regularization on the weight vectors of the last fully-connected layer and the last hidden activation. Assume $s_c(x) = \hat{y}_c$ is the c-th output of the model for the c-th class. Then we obtain the calibrated loss with enforced margins:

$$L^{CAL}((x, y); s) = -\log \frac{e^{\hat{y}_c - \Delta_c}}{e^{\hat{y}_c - \Delta_c} + \sum_{c \neq y} e^{\hat{y}_c}}, \tag{13}$$

where $\Delta_c = \frac{H}{N_c^{1/4}}$, for $c \in \{1, \cdots, C\}$. For balanced datasets, Δ_c depends on constant C. Conversely, in highly skewed datasets, the value of Δ_c is determined by the variability in label distribution. The calibrated loss from Eq. (13) is implemented during client updates to minimize bias.

V. Experiment

Our study includes two key experiments designed to validate (i) the calibration method's effectiveness in managing highly skewed data and (ii) the robustness of our approach against leading FL models. The first experiment assesses performance disparities pre- and post-calibration application. Second, the second experiment benchmarks our proposed method against established models. Details of the experimental setup are outlined in Section V-A, with findings from the calibration and robustness results presented in Sections V-B1 and V-B2, respectively.

A. Experimental Setup

1) Dataset: To evaluate the proposed calibration method's performance, our experiments were conducted using three benchmark datasets: MNIST, CIFAR-10, and the Adult Income dataset. The initial experiment utilized all three datasets, while the second experiment focused on MNIST and CIFAR-10.

- MNIST [33]: A collection of 28x28 pixel grayscale images of handwritten digits, comprising 10 classes. The model uses 2352 features for classification into 10 output classes.
- CIFAR-10 [34]: This dataset consists of 60,000 32x32 color images in 10 different classes, with 6,000 images per class.
- Adult Income [35]: A tabular dataset with demographic attributes to predict whether an individual earns more than $50K/year, effectively a binary classification task. The model inputs consist of 14 features to determine 1 output class for the binary classification.

2) Data Partitioning Strategy: To mirror real-world conditions in federated learning, we partitioned data among clients using a Dirichlet distribution $D(\alpha)$. The hyperparameter α controls the level of data imbalance across clients. The Dirichlet distribution is defined as:

$$f(p_1, \ldots, p_C; \alpha_1, \ldots, \alpha_C) = \frac{1}{B(\alpha)} \prod_{i=1}^{C} p_i^{\alpha_i - 1} I(p \in S), \tag{14}$$

where $S = \{p_i \in [0, 1], \sum_{i=1}^{C} p_i = 1\}$, $B(\alpha) = \frac{\prod_{i=1}^{C} \Gamma(\alpha_i)}{\Gamma(\alpha_0)}$, and $\alpha_0 = \sum_{i=1}^{C} \alpha_i$. This allows for varying degrees of data skewness by adjusting α; higher values lead to uniform distributions, while lower values result in higher skewness. For each label c, client data distributions $P_c \sim \text{Dir}(\alpha)$ are sampled and assigned to clients k. This models realistic scenarios where client data may be majority, minority, or even missing.

Fig. 2 demonstrates our partitioning strategy on MNIST and Adult Income datasets under varying α values, showing the range from nearly uniform ($\alpha = 10000$) to highly imbalanced ($\alpha = 0.05$).

3) Implementation: We used a Multi-Layer Perceptron (MLP) for MNIST and Adult Income, and a Convolutional Neural Network (CNN) for CIFAR-10, each suited to their respective datasets. More specifications are in I.

Clients trained locally on their partitions for 3 epochs in all experiments. The server aggregated client updates over 10 global epochs for MNIST and Adult Income and 15 for CIFAR-10. Learning rates were 0.001 for MNIST and Adult Income, and 0.0003 for CIFAR-10, with a momentum factor of 0.9 across all experiments. Optimizers were Stochastic Gradient Descent (SGD) for MNIST and Adam for CIFAR-10 and Adult Income. Data split ratios were 0.3 for MNIST and 0.2 for CIFAR-10 and Adult Income, defining the data proportion each client managed.

TABLE I
Neural Network Model Architecture

Model	Architecture
MLP	Input Hidden-512 + ReLu Hidden-256 + ReLu Hidden-128 + ReLu Hidden-64 + ReLu Output
CNN	conv3-128 + maxpool conv3-256 + maxpool 2 x conv3-512 conv3-256 + maxpool FC-512 FC-256 FC-128 FC-10 + ReLu

4) Baseline: Our approach is first compared to the conventional Federated Averaging (FedAvg) algorithm [1], which serves as the baseline for this study. The initial experiment examines the performance of FedAvg before and after applying our calibration technique, highlighting its direct impact. In the second experiment, we expand the comparison to include other advanced federated learning methods such as FedLC [25], MOON [36], and FedProx [7].

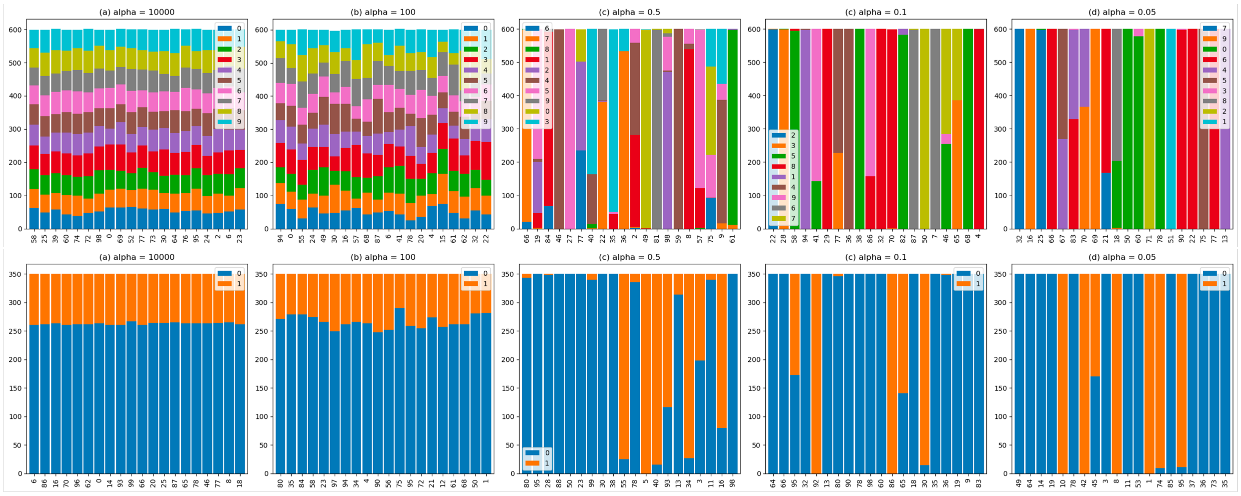


Fig. 2. Client data distribution under varying α Dirichlet distribution for MNIST (top) and Adult Income (bottom) datasets.

5) Evaluation: We reported accuracy and loss as key metrics to assess our global model's performance. Specifically, we applied multi-class cross-entropy loss for MNIST and CIFAR-10 datasets, and binary cross-entropy loss for the Adult Income dataset. For a detailed analysis, we also generated t-SNE plots and confusion matrices for clearer visualization. The following are the metrics used for evaluation:

$$\text{Accuracy} = \frac{\sum_{i=1}^{N} \mathbf{1}(\hat{y}_i = y_i)}{N}, \tag{15}$$

where N is the total number of samples, y_i is the true label for the i-th sample, $\hat{y}_i$ is the predicted label for the i-th sample, and $\mathbf{1}(\hat{y}_i = y_i)$ is an indicator function that equals 1 if $\hat{y}_i = y_i$ and 0 otherwise.

$$L_{\text{multi-class}} = -\sum_{i=1}^{N}\sum_{c=1}^{C} y_{ic}\log(\hat{y}_{ic}), \tag{16}$$

where N is the number of samples, C is the number of classes, y_{ic} is a binary indicator (0 or 1) if class c is the correct classification for sample i, and $\hat{y}_{ic}$ is the predicted probability that sample i belongs to class c.

$$L_{\text{binary}} = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i\log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right), \tag{17}$$

where N is the number of samples, y_i is the true label for the i-th sample (0 or 1), and $\hat{y}_i$ is the predicted probability that sample i belongs to the positive class (label 1).

B. Experimental Result

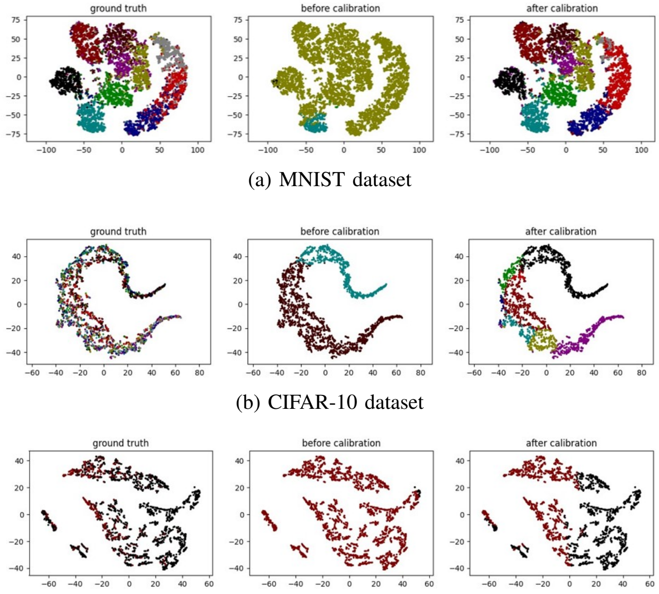
1) Calibration Experiment: In the calibration experiment, we evaluated our method using the MNIST, CIFAR10, and Adult Income datasets, each with varying skewness levels. The goal was to assess the impact of our calibration on model performance by comparing results from the conventional FedAvg algorithm to our proposed method. As shown in Table II, our calibration significantly improves accuracy, especially in highly imbalanced data scenarios. For MNIST, the t-SNE plots (Figure 3a) reveal well-defined clusters post-calibration, with the most significant accuracy improvement at $\alpha = 0.1$, increasing from 49.08% to 87.64%. Similarly, CIFAR-10 results (Figure 3b) show distinct clusters post-calibration, with a 13.04% accuracy improvement at $\alpha = 0.1$. However, the performance gains diminished as the skewness decreased. The Adult Income dataset (Figure 3c) exhibited the most notable improvement at $\alpha = 0.05$, with a dramatic accuracy boost of 43.83%, significantly reducing misclassifications. These results, highlighted by confusion matrices in Figure 4, confirm that our calibration method enhances model performance under varying data skewness, particularly in highly imbalanced scenarios.

TABLE II
COMPARISON OF ACCURACY BETWEEN FEDAVG AND OUR PROPOSED METHOD. IMPROVEMENT IS HIGHLIGHTED IN RED. LOWER SKEWNESS INDICATES MORE IMBALANCED DATA.

Dataset	Method	Degree of Skewness				
		0.05	0.1	0.3	0.5	0.7
MNIST	FedAvg	40.11	49.08	84.75	87.68	87.86
	Ours	74.16	87.64	90.80	88.23	92.24
	Difference	+34.05	+38.56	+6.05	+1.17	+3.97
CIFAR-10	FedAvg	7.25	10.00	48.71	69.26	71.38
	Ours	19.17	23.04	54.27	70.19	70.31
	Difference	+11.92	+13.04	+5.56	+0.93	-1.07
Adult Income	FedAvg	24.64	52.94	50.09	50.81	66.25
	Ours	68.47	67.66	64.37	70.31	73.34
	Difference	+43.83	+14.72	+14.28	+19.50	+7.09

In conclusion, the calibration experiment validates that our approach enhances model performance under varying data skewness. The calibration effect diminishes as the data distribution becomes balanced, but our research primarily addresses scenarios with significant dataset imbalances.

2) Robustness Experiment: In this experiment, detailed in Tables III and IV, we evaluated the robustness of our FL-FCR method against other leading FL models: FedCL [25], MOON [36], and FedProx [7] under highly imbalanced settings ($\alpha \in [0.05, 0.1, 0.2, 0.3]$). The results show that our method consistently outperforms these benchmarks, especially

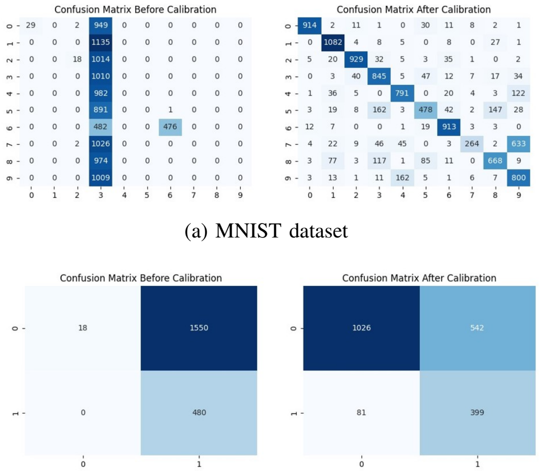


(a) MNIST dataset

(b) CIFAR-10 dataset

(c) Adult Income dataset

Fig. 3. The t-SNE plots illustrate the improved model performance for MNIST, CIFAR-10, and Adult Income datasets at $\alpha = 0.05$.



Confusion Matrix Before Calibration

	0	1	2	3	4	5	6	7	8	9
0	29	0	2	949	0	0	0	0	0	0
1	0	0	0	1135	0	0	0	0	0	0
2	0	0	18	1014	0	0	0	0	0	0
3	0	0	0	1010	0	0	0	0	0	0
4	0	0	0	982	0	0	0	0	0	0
5	0	0	0	891	0	0	1	0	0	0
6	0	0	0	482	0	0	476	0	0	0
7	0	0	2	1026	0	0	0	0	0	0
8	0	0	0	974	0	0	0	0	0	0
9	0	0	0	1009	0	0	0	0	0	0

Confusion Matrix After Calibration

	0	1	2	3	4	5	6	7	8	9
0	914	2	11	1	0	30	11	8	2	1
1	0	1082	4	8	5	0	8	0	27	1
2	5	20	929	32	5	3	35	1	0	2
3	0	3	40	845	5	47	12	7	17	34
4	1	36	5	0	791	0	20	4	3	122
5	3	19	8	162	3	478	42	2	147	28
6	12	7	0	0	1	19	913	3	3	0
7	4	22	9	46	45	0	3	264	2	633
8	3	77	3	117	1	85	11	0	668	9
9	3	13	1	11	162	5	1	6	7	800

(a) MNIST dataset

Confusion Matrix Before Calibration

	0	1
0	18	1550
1	0	480

Confusion Matrix After Calibration

	0	1
0	1026	542
1	81	399

(b) Adult Income dataset

Fig. 4. Confusion matrices for the MNIST and Adult Income datasets at $\alpha = 0.05$, illustrating classification performance before and after applying FL-FCR. (a) MNIST shows improved accuracy and clearer class separation. (b) Adult Income indicates fewer false negatives and more balanced classification, enhancing overall predictive accuracy.

under extreme skewness.

For the MNIST dataset, our model achieved an accuracy of 65.34% at $\alpha = 0.05$ (highly skewed) and maintained an average accuracy of 78.04%. Figure 5 shows that our method consistently outperforms FedCL, MOON, and FedProx across different skewness levels, indicating robustness. For the CIFAR10 dataset, our method demonstrated substantial effectiveness, particularly at $\alpha = 0.05$, with an accuracy of 19.81%, higher than the benchmarks. The overall average accuracy of 39.80% confirms its robustness in addressing data imbalance.

Other state-of-the-art methods struggle with imbalanced data due to their design focus. MOON [36] enhances model consistency across clients using a contrastive loss but lacks mechanisms to address skewed data distributions. FedProx [7] introduces a proximal term to the local training objective to improve convergence under heterogeneous data. However, it primarily targets system-level heterogeneity (such as varying computational capabilities and network conditions) rather than statistical heterogeneity (data imbalance). This proximal term stabilizes the training process by preventing local updates from deviating too far from the global model, but it does not address the uneven representation of classes in the data. These findings are significant for real-world applications where data is often highly imbalanced. Our method ensures robust and consistent performance across various skewness levels, making it suitable for diverse federated learning scenarios.

TABLE III

ACCURACY ACROSS DIFFERENT LEVELS OF SKEWNESS FOR MNIST AND CIFAR-10 DATASETS. RED INDICATES THE BEST PERFORMANCE.

Accuracy		0.05	0.1	0.2	0.3	Avg.
MNIST	FedCL [25]	33.14	48.29	78.85	88.16	62.11
	MOON [36]	43.44	66.55	77.64	90.81	69.61
	FedProx [7]	46.25	71.59	73.7	83.93	68.87
	Ours	65.34	73.45	81.2	92.15	78.04
CIFAR10	FedCL [25]	14.8	22.97	47.11	64.38	37.32
	MOON [36]	10.51	24.44	59.57	61.94	39.12
	FedProx [7]	14.85	18.22	51.81	66.53	37.85
	Ours	19.81	27.85	50.73	60.78	39.80

TABLE IV

LOSS ACROSS DIFFERENT LEVELS OF SKEWNESS FOR MNIST AND CIFAR-10 DATASETS. RED INDICATES THE BEST PERFORMANCE.

Loss		0.05	0.1	0.2	0.3	Avg.
MNIST	FedCL [25]	0.0455	0.0222	0.0111	0.0067	0.0214
	MOON [36]	0.0286	0.0178	0.0118	0.0058	0.0160
	FedProx [7]	0.0224	0.0175	0.0129	0.0105	0.0158
	Ours	0.0205	0.0157	0.0103	0.0095	0.0140
CIFAR10	FedCL [25]	0.0356	0.0343	0.0257	0.0193	0.0287
	MOON [36]	0.0364	0.0328	0.0209	0.0188	0.0272
	FedProx [7]	0.0357	0.0348	0.0251	0.0160	0.0279
	Ours	0.0349	0.0312	0.0224	0.0227	0.0278

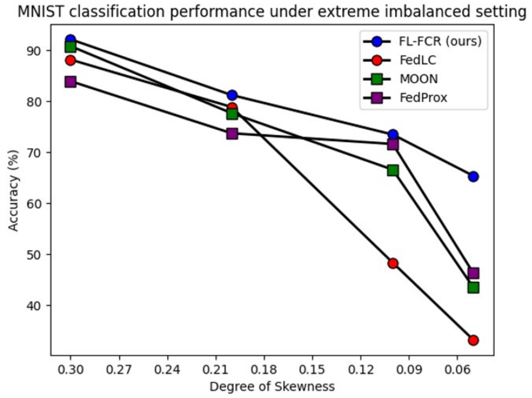


Fig. 5. MNIST classification performance under extreme imbalanced setting.

VI. Conclusion

Our Federated Learning with Feature Calibration and Client Resampling (FL-FCR) approach introduces two key techniques to enhance data privacy and model performance in federated learning. By using calibration statistics for dataset resampling and replacing the conventional SGD loss function with a calibrated loss function, our method effectively addresses real-world imbalanced datasets.

Empirical evaluations show that FL-FCR outperforms FedAvg and other leading FL methods in accuracy and loss reduction while improving data privacy and security. Calibration weighting and resampling provide an extra layer of privacy protection by minimizing the impact of outliers and malicious data points.

Although our method requires additional computational overhead due to server-side retraining, future research could explore optimized retraining strategies or more efficient calibration techniques. Scalability concerns for extremely large datasets or numerous clients also warrant further investigation.

Our work contributes to the fields of data mining, information forensics, and security by demonstrating enhanced robustness and security in federated learning, especially with imbalanced data. Future research should focus on improving scalability and efficiency and exploring applications in various security-critical domains. Additionally, integrating our approach with other privacy-preserving techniques, such as differential privacy or homomorphic encryption, could further enhance the security and applicability of FL systems.

Appendix A
Appendix: Global Sample Covariance

Assuming $N_{c,k} \geq 2$, $\mathrm{cov}_{c,k}$ of class c in client k (local covariance) can be derived as:

$$\begin{aligned}\mathrm{cov}_{c,k} &= \frac{1}{N_{c,k}-1}\Bigg[\sum_{i=1}^{N_{c,k}} v_{c,k,i}v_{c,k,i}^\top - \sum_{i=1}^{N_{c,k}} v_{c,k,i}\mu_{c,k}^\top \\ &\quad - \sum_{i=1}^{N_{c,k}} \mu_{c,k}v_{c,k,i}^\top + \sum_{i=1}^{N_{c,k}} \mu_{c,k}\mu_{c,k}^\top\Bigg] \\ &= \frac{1}{N_{c,k}-1}\Bigg[\sum_{i=1}^{N_{c,k}} v_{c,k,i}v_{c,k,i}^\top - N_{c,k}\mu_{c,k}\mu_{c,k}^\top \\ &\quad - N_{c,k}\mu_{c,k}\mu_{c,k}^\top + N_{c,k}\mu_{c,k}\mu_{c,k}^\top\Big] \\ &= \frac{1}{N_{c,k}-1}\sum_{i=1}^{N_{c,k}} v_{c,k,i}v_{c,k,i}^\top - \frac{N_{c,k}}{N_{c,k}-1}\mu_{c,k}\mu_{c,k}^\top\end{aligned} \tag{18}$$

After rearranging, $\mathrm{cov}_{c,k}$ can also be written as:

$$(N_{c,k}-1)\mathrm{cov}_{c,k} = \sum_{i=1}^{N_{c,k}} v_{c,k,i}v_{c,k,i}^\top - N_{c,k}\mu_{c,k}\mu_{c,k}^\top \tag{19}$$

Lastly, global sample covariance cov_c can be derived by the results obtained from (3) and (4):

$$\mathrm{cov}_c = \frac{1}{N_c-1}\sum_{k=1}^{K}\sum_{i=1}^{N_{c,k}}(v_{c,k,i}-\mu_c)(v_{c,k,i}-\mu_c)^\top \tag{20}$$

$$= \frac{1}{N_c-1}\sum_{k=1}^{K}\sum_{i=1}^{N_{c,k}} v_{c,k,i}v_{c,k,i}^\top - \frac{N_c}{N_c-1}\mu_c\mu_c^\top \tag{21}$$

$$= \sum_{k=1}^{K}\frac{1}{N_c-1}\left[(N_{c,k}-1)\mathrm{cov}_{c,k} + N_{c,k}\mu_{c,k}\mu_{c,k}^\top\right] \tag{22}$$

$$- \frac{N_c}{N_c-1}\mu_c\mu_c^\top \tag{23}$$

$$= \sum_{k=1}^{K}\frac{N_{c,k}-1}{N_c-1}\mathrm{cov}_{c,k} + \sum_{k=1}^{K}\frac{N_{c,k}}{N_c-1}\mu_{c,k}\mu_{c,k}^\top \tag{24}$$

$$- \frac{N_c}{N_c-1}\mu_c\mu_c^\top. \tag{25}$$

The first equality holds by the definition of sample covariance, the second equality holds due to (3), and the third equality is obtained from (4).

Appendix B
Appendix: CALIBRATED LOSS FUNCTION

In this appendix, we provide the theoretical foundation and derivation of the calibrated loss function designed to address imbalanced datasets in classification tasks.

A. Softmax Cross-Entropy Loss Derivation

For input x with true label y, let $s(x)$ denote the score vector. The predicted label $\hat{y}$ is:

$$\hat{y} = \text{arg max } s(x)$$

The softmax cross-entropy loss $L(y, \hat{y})$ is:

$$L(y,\hat{y}) = L(y, s(x)) = -\log\frac{e^{s_y(x)}}{\sum_{y'\in[C]} e^{s_{y'}(x)}}$$

Simplified as:

$$L(y, s(x)) = \log\left[\sum_{y'\in[C]} e^{s_{y'}(x)}\right] - s_y(x)$$

Or:

$$L(y, s(x)) = \log\left[1 + \sum_{y'\neq y} e^{s_{y'}(x)-s_y(x)}\right]$$

B. Derivation of Equation (9)

Error bound for class c:

$$L_c[f]\frac{1}{\gamma_c}\sqrt{\frac{C(\mathcal{F})}{N_c}} + \frac{\log N}{\sqrt{N_c}} \tag{26}$$

Balanced error bound:

$$L_{\text{bal}}[f]\frac{1}{C}\sum_{c=1}^{C}\left(\frac{1}{\gamma_c}\sqrt{\frac{C(\mathcal{F})}{N_c}} + \frac{\log N}{\sqrt{N_c}}\right) \tag{27}$$

For $C = 2, [C] = \{p, q\}$:

$$L_{\text{bal}}[f]\frac{1}{2}\left(\frac{1}{\gamma_p}\sqrt{\frac{C(\mathcal{F})}{N_p}} + \frac{\log N}{\sqrt{N_p}} + \frac{1}{\gamma_q}\sqrt{\frac{C(\mathcal{F})}{N_q}} + \frac{\log N}{\sqrt{N_q}}\right) \tag{28}$$

Ignoring logarithmic terms:

$$L_{\text{bal}}[f] \approx \frac{1}{2}\left(\frac{1}{\gamma_p}\sqrt{\frac{C(\mathcal{F})}{N_p}} + \frac{1}{\gamma_q}\sqrt{\frac{C(\mathcal{F})}{N_q}}\right) \tag{29}$$

For binary classification:

$$e^{\text{Cal}}\frac{1}{\gamma_p\sqrt{N_p}} + \frac{1}{\gamma_q\sqrt{N_q}} \tag{30}$$

C. Derivation of Equation (10)

Perturbed margins $\gamma'_p = \gamma_p - \delta$ and $\gamma'_q = \gamma_q + \delta$:

$$e_{\delta}^{\text{Cal}}\frac{1}{(\gamma_p - \delta)\sqrt{N_p}} + \frac{1}{(\gamma_q + \delta)\sqrt{N_q}}$$

Optimal margins must satisfy:

$$\frac{1}{\gamma_p\sqrt{N_p}} + \frac{1}{\gamma_q\sqrt{N_q}} \geq \frac{1}{(\gamma_p - \delta)\sqrt{N_p}} + \frac{1}{(\gamma_q + \delta)\sqrt{N_q}}$$

D. Derivation of Equation (11)

Fixed sum of margins $\gamma_p + \gamma_q = \beta$:

$$\gamma_q = \beta - \gamma_p$$

Minimize:

$$\frac{1}{\gamma_p\sqrt{N_p}} + \frac{1}{(\beta - \gamma_p)\sqrt{N_q}}$$

Set derivative to zero:

$$-\frac{1}{\gamma_p^2\sqrt{N_p}} + \frac{1}{(\beta - \gamma_p)^2\sqrt{N_q}} = 0$$

Solve for γ_p:

$$\gamma_p = \frac{\beta\sqrt[4]{N_p}}{\sqrt[4]{N_p} + \sqrt[4]{N_q}}$$

And for γ_q:

$$\gamma_q = \frac{\beta\sqrt[4]{N_q}}{\sqrt[4]{N_p} + \sqrt[4]{N_q}}$$

E. Derivation of Equation (12)

To extend the result to a multi-class setting, consider the balanced error bound:

$$L_{\text{bal}}[f]\frac{1}{C}\sum_{c=1}^{C}\left(\frac{1}{\gamma_c}\sqrt{\frac{C(\mathcal{F})}{N_c}}\right)$$

For a fixed sum of margins, we let:

$$\gamma_c = \frac{H}{N_c^{1/4}}$$

where H is a hyper-parameter to be tuned and N_c is the number of samples in class c.

Substituting $\gamma_c = \frac{H}{N_c^{1/4}}$ into the error bound, we get:

$$L_{\text{bal}}[f]\frac{1}{C}\sum_{c=1}^{C}\left(\frac{1}{\frac{H}{N_c^{1/4}}}\sqrt{\frac{C(\mathcal{F})}{N_c}}\right)$$

Simplifying:

$$L_{\text{bal}}[f]\frac{1}{C}\sum_{c=1}^{C}\left(\frac{N_c^{1/4}}{H}\sqrt{\frac{C(\mathcal{F})}{N_c}}\right)$$

$$L_{\text{bal}}[f]\frac{1}{C}\sum_{c=1}^{C}\left(\frac{N_c^{1/4}}{H}\sqrt{C(\mathcal{F})}N_c^{-1/2}\right)$$

$$L_{\text{bal}}[f]\frac{1}{C}\sum_{c=1}^{C}\left(\frac{\sqrt{C(\mathcal{F})}}{H}N_c^{-1/4}\right)$$

By using the relationship between the margins and sample sizes, the optimal margin for each class γ_c ensures that the error bound is minimized. Thus, the optimal margin Δ_c for each class c is given by:

$$\Delta_c = \frac{H}{N_c^{1/4}}$$

F. Derivation of Equation (13)

The traditional softmax loss function for an input x and true label y is:

$$L_{softmax} = -\log\frac{e^{\hat{y}_c}}{\sum_{j=1}^{C} e^{\hat{y}_j}}$$

where $\hat{y}_c$ is the output (logit) corresponding to the true class c.

AM-Softmax [32] introduces an additive margin m to the logits. The AM-Softmax loss is:

$$L_{AMS} = -\frac{1}{n}\sum_{i=1}^{n}\log\frac{e^{s(\cos\theta_{y_i} - m)}}{e^{s(\cos\theta_{y_i} - m)} + \sum_{j \neq y_i} e^{s\cos\theta_j}}$$

where s is a scaling factor, and θ_{y_i} is the angle between the weight vector and the feature vector of the i-th sample for the true class y_i.

To address class imbalance, we introduce a calibrated margin Δ_c for each class c. The calibrated margin is defined as:

$$\Delta_c = \frac{H}{N_c^{1/4}}$$

where H is a constant and N_c is the number of samples in class c.

Incorporating the calibrated margin into the softmax loss, we get the calibrated loss function:

$$L^{CAL}((x, y); s) = -\log\frac{e^{\hat{y}_c - \Delta_c}}{e^{\hat{y}_c - \Delta_c} + \sum_{c \neq y} e^{\hat{y}_c}}$$

This loss function adjusts the logits based on the class distribution, reducing the bias towards majority classes. By applying the calibrated margin, the loss function effectively penalizes classes proportionally to their prevalence, ensuring fair treatment across all classes.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1273–1282.

[2] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[3] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[4] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang *et al.*, "Large scale distributed deep networks," *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[5] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, "Accurate, large minibatch sgd: Training imagenet in 1 hour," *arXiv preprint arXiv:1706.02677*, 2017.

[6] W. J. Reed, "The pareto, zipf and other power laws," *Economics Letters*, vol. 74, no. 1, pp. 15–19, 2001.

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

[8] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582*, 2018.

[9] T.-M. H. Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," *arXiv preprint arXiv:1909.06335*, 2019.

[10] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249–259, 2018.

[11] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020.

[12] Z. Ma, J. Ma, Y. Miao, Y. Li, and R. H. Deng, "Shieldfl: Mitigating model poisoning attacks in privacy-preserving federated learning," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 1639–1654, 2022.

[13] X. Liu, H. Li, G. Xu, Z. Chen, X. Huang, and R. Lu, "Privacy-enhanced federated learning against poisoning adversaries," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 4574–4588, 2021.

[14] J. Zhao, H. Zhu, F. Wang, R. Lu, Z. Liu, and H. Li, "Pvd-fl: A privacy-preserving and verifiable decentralized federated learning framework," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 2059–2073, 2022.

[15] K. Wei, J. Li, C. Ma, M. Ding, W. Chen, J. Wu, M. Tao, and H. V. Poor, "Personalized federated learning with differential privacy and convergence guarantee," *IEEE Transactions on Information Forensics and Security*, 2023.

[16] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5132–5143.

[17] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7611–7623, 2020.

[18] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated learning with matched averaging," *arXiv preprint arXiv:2002.06440*, 2020.

[19] J.-C. Deville and C.-E. Sarndal, "Calibration estimators in survey sampling," *Journal of the American Statistical Association*, pp. 376–382, 1992.

[20] R. Valliant, J. A. Dever, and F. Kreuter, *Practical tools for designing and weighting survey samples*. Springer, 2013, vol. 1.

[21] R. J. Little and S. Vartivarian, "Does weighting for nonresponse increase the variance of survey means?" *Survey Methodology*, vol. 31, no. 2, p. 161, 2005.

[22] K. E. Irimata, Y. He, V. L. Parsons, H.-C. Shin, and G. Zhang, "Calibration weighting methods for the national center for health statistics research and development survey." *Vital and Health Statistics. Ser. 1, Programs and Collection Procedures*, no. 87, pp. 1–23, 2023.

[23] T. Chang and P. S. Kott, "Using calibration weighting to adjust for nonresponse under a plausible model," *Biometrika*, vol. 95, no. 3, pp. 555–571, 2008.

[24] S. F. Tett, K. Yamazaki, M. J. Mineter, C. Cartis, and N. Eizenberg, "Calibrating climate models using inverse methods: case studies with hadam3, hadam3p and hadcm3," *Geoscientific Model Development*, vol. 10, no. 9, pp. 3567–3589, 2017.

[25] J. Zhang, Z. Li, B. Li, J. Xu, S. Wu, S. Ding, and C. Wu, "Federated learning with label distribution skew via logits calibration," in *International Conference on Machine Learning*. PMLR, 2022, pp. 26 311–26 329.

[26] M. Luo, F. Chen, D. Hu, Y. Zhang, J. Liang, and J. Feng, "No fear of heterogeneity: Classifier calibration for federated learning with non-iid data," *Advances in Neural Information Processing Systems*, vol. 34, pp. 5972–5984, 2021.

[27] X. Shang, Y. Lu, Y.-m. Cheung, and H. Wang, "Fedic: Federated learning on non-iid and long-tailed data via calibrated distillation," in *2022 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2022, pp. 1–6.

[28] Z. Li, X. Shang, R. He, T. Lin, and C. Wu, "No fear of classifier biases: Neural collapse inspired federated learning with synthetic and fixed classifier," *arXiv preprint arXiv:2303.10058*, 2023.

[29] A. K. Menon, S. Jayasumana, A. S. Rawat, H. Jain, A. Veit, and S. Kumar, "Long-tail learning via logit adjustment," *arXiv preprint arXiv:2007.07314*, 2020.

[30] K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma, "Learning imbalanced datasets with label-distribution-aware margin loss," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[31] S. M. Kakade, K. Sridharan, and A. Tewari, "On the complexity of linear prediction: Risk bounds, margin bounds, and regularization," *Advances in Neural Information Processing Systems*, vol. 21, 2008.

[32] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.

[33] Y. LeCun, C. Cortes, and C. Burges, "The mnist database of handwritten digits," http://yann.lecun.com/exdb/mnist/, 2010.

[34] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," https://www.cs.toronto.edu/ kriz/learning-features-2009-TR.pdf, Citeseer, Tech. Rep., 2009, accessed: insert-access-date-here.

[35] D. Dua and C. Graff, "UCI machine learning repository," http://archive.ics.uci.edu/ml, 2017, university of California, Irvine, School of Information and Computer Sciences.

[36] Q. Li, B. He, and D. Song, "Model-contrastive federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 713–10 722.