electronics

Article

A Federated-Learning Algorithm Based on Client Sampling and Gradient Projection for the Smart Grid

Ruifeng Zhao [1], Jiangang Lu [1], Zewei Liu [2,*], Tianqi Wang [2], Wenxin Guo [1], Tian Lan [1] and Chunqiang Hu [2]

[1] Electric Power Dispatching and Control Center of Guangdong Power Grid Co., Ltd., Guangzhou 510062, China; ruifzhao@126.com (R.Z.); lujiangang@gddd.csg.cn (J.L.); guowenxin@gddd.csg.cn (W.G.); lantian@gddd.csg.cn (T.L.)

[2] School of Big Data and Software Engineering, Chongqing University, Chongqing 401331, China; tqwang@cqu.edu.cn (T.W.); chu@cqu.edu.cn (C.H.)

* Correspondence: zwliu@cqu.edu.cn

Abstract: Federated learning (FL) is a machine-learning framework that effectively addresses privacy concerns. It harnesses fragmented data from devices across the globe for model training and optimization while strictly adhering to user privacy protection and regulatory compliance. This framework holds immense potential for widespread applications in the smart-grid domain. Through FL, power companies can collaborate to train smart-grid models without revealing users' electricity consumption data, thus safeguarding their privacy. However, the data collected by clients often exhibits heterogeneity, which can lead to biases towards certain data features during the model-training process, therefore affecting the fairness and performance of the model. To tackle the fairness challenges that emerge during the federated-learning process in smart grids, this paper introduces FedCSGP, a novel federated-learning approach that incorporates client sampling and gradient projection. The main idea of FedCSGP is to categorize the causes of unfairness in federated learning into two parts: internal conflicts and external conflicts. Among them, the client-sampling strategy is used to resolve external conflicts, while the gradient-projection strategy is employed to address internal conflicts. By tackling both aspects, FederCSGP aims to enhance the fairness of the federated-learning model while ensuring the accuracy of the global model. The experimental results demonstrate that the proposed method significantly improves the accuracy of poorly performing clients in smart-grid scenarios with lower communication costs, therefore enhancing the fairness of the federated-learning algorithm.

Keywords: federated learning; fairness; communication costs

check for updates

Citation: Zhao, R.; Lu, J.; Liu, Z.; Wang, T.; Guo, W.; Lan, T.; Hu, C. A Federated-Learning Algorithm Based on Client Sampling and Gradient Projection for the Smart Grid. *Electronics* **2024**, *13*, 2023. https://doi.org/10.3390/electronics13112023

Academic Editor: Ahmed Abu-Siada

Received: 16 April 2024
Revised: 17 May 2024
Accepted: 20 May 2024
Published: 22 May 2024

1. Introduction

The rapid development of high-tech technologies such as big data, artificial intelligence, the Internet of Things (IoT), digitization, and intelligence has become the daily work and lifestyle of people worldwide [1]. However, various industries are currently grappling with challenges related to fragmented data, limited data sharing, and data silos, which pose significant obstacles to the further advancement of high-tech technologies. Additionally, as the frequency of data sharing and utilization increases, concerns regarding data privacy and security have become more pronounced. To address these concerns, countries around the world are continuously enacting and refining laws and regulations pertaining to data security and privacy protection. Examples include the General Data Protection Regulation (GDPR) [2] in the European Union, the California Consumer Privacy Act (CCPA) [3] in the United States, and China's Personal Information Protection Specification (GB/T 35273) [4]. Protecting data privacy and security has become a hot topic of interest for global researchers and businesses. Consequently, researchers are actively exploring new methods that can leverage fragmented data for machine learning while ensuring data privacy and security. It is in response to this demand that federated learning (FL) has emerged.

Unlike traditional machine learning, in federated learning, each device updates the model parameters based on local data and then uploads the model parameters to a central server for aggregation to obtain the global model [5,6]. This approach not only protects user data privacy but also eliminates data barriers between enterprises, addressing the issue of data silos. Currently, federated learning finds good applications in smart grids. First, smart grids involve a large amount of electricity consumption data from users, which raises privacy concerns. Federated learning enables model training without exposing raw data, thus protecting user privacy. Second, data in smart grids is often distributed across different geographical locations or organizations. Federated learning facilitates model training on distributed data, eliminating the necessity to centralize all the data in a single location. Moreover, given that smart grids are vital systems pertaining to energy supply and stability, security emerges as a crucial aspect. Federated learning serves to minimize the potential risk of data breaches and attacks [7].

Indeed, FL, as a unique distributed machine-learning approach, presents distinct challenges related to fairness due to its characteristics of device heterogeneity, data heterogeneity, and its unique model-training paradigm. Ignoring fairness can lead to discrimination or bias towards certain participants or groups, giving rise to problems of inequality and discrimination. Addressing the fairness issue in federated learning helps improve the performance of the global model, making its application more stable and reliable among all participants. It also encourages active participation in federated-learning training, fosters multi-party collaboration, promotes the sustainable development of federated learning, and ultimately helps break down data barriers and resolve the problem of data silos. Absolutely, when researching and implementing federated learning, it is crucial to pay close attention to its potential fairness issues [8].

Agnostic-Fair [9] aims to mitigate disparities and biases in models across different subgroups by incorporating fairness constraints. FL+HC (hierarchical clustering) [10] aims to address the challenges of non-IID (non-independent and identically distributed) data distribution and improve the global model's performance. Avishek Ghosh et al. [11] consider heterogeneous data and device characteristics through hierarchical clustering and reduce the variation and bias of the model across subgroups by introducing fairness constraints. Smith et al. [12] aims to generate different local models for different clients based on a global model trained. However, it is important to acknowledge that these approaches may introduce additional computational and communication overhead.

While existing research efforts have taken potential fairness issues into account in implementing federated learning, little consideration has been given to cost-effectively addressing client gradient conflicts. In the process of federated learning, various clients (or participants) train models using local data and upload their gradient updates to a central server for aggregation. Significant differences in data distribution, quantity, and quality among clients can lead to gradient conflicts during global model aggregation. This conflict inherently reflects unfairness in the model optimization process, as the gradients from certain clients may have a disproportionate impact on the global model due to their data volume or quality advantages, potentially sacrificing the interests of other clients. To ensure fairness, it is crucial to devise reasonable aggregation algorithms and optimization strategies that balance the contributions of different clients and minimize the impact of gradient conflicts on the performance of the global model.

To achieve fairness in the global model while ensuring high accuracy and reducing communication costs during federated learning in smart grids, we introduce a federated-learning algorithm based on client sampling and gradient projection, referred to as FedC-SGP (Federated Learning with Client Sampling and Gradient Projection). This algorithm divides the training process of the global model into two stages: pre-training and formal training. In the pre-training stage, clients with similar data features are clustered together using a hierarchical clustering method. In the formal training stage, the central server selects clients from each cluster to participate in the training. The algorithm checks for gradient conflicts among clients from different clusters and mitigates conflicts through gradient projection. This approach improves the representativeness of participating clients in each round, alleviates conflicts among different clients, enhances the fairness of the global model, and reduces communication costs.

The primary contributions of this paper can be summarized as follows:

- We propose a federated-learning algorithm tailored for smart grids, which is based on client-sampling strategies and gradient-projection techniques. This algorithm divides the model-training process in federated learning into two stages: pre-training and formal training.
- In the pre-training stage, a client-sampling strategy is employed to address external conflicts. In the formal training stage, the gradient-projection algorithm is used to handle internal conflicts.
- By resolving internal and external conflicts during federated-learning training, the proposed algorithm enhances the fairness of federated-learning models while ensuring global model accuracy and reducing communication costs.

The remainder of this paper is organized as follows. Section 2 introduces related works. Section 3 shows the overall detailed flow of FedCSGP. The performance analyses are adduced in Sections 4. Finally, conclusions are drawn in Section 5.

2. Related Work

The fairness issue in federated learning was first proposed by Kairouz et al. [13], and current research in both domestic and international fields can be summarized into the following three categories for improving the fairness of federated learning:

2.1. Based on Client Weight Allocation Strategies

These methods generally increase computation costs and communication overheads. For example, Kairouz et al. [13] summarized existing fairness measurement methods and proposed improving fairness by adjusting weight updating strategies. Mohri et al. [14] proposed the Agnostic Federated-Learning (AFL) algorithm based on Min-Max optimization, which aims to balance the quality of local model updates and the consistency of global models in each iteration, with a focus on the poorest-performing devices. Wang et al. [15] proposed the FedNova algorithm. This algorithm utilizes efficient local steps and aggregated weights to adjust the progress of each client, enabling it to converge to a stable point of the true objective function and produce accurate results. FedNova retains fast error convergence while addressing the issue of objective inconsistency. However, the algorithm lacks adaptive optimization and is not suitable for gossip-based training methods. Hu et al. [16] proposed the FedMGDA+ algorithm, which modifies the model gradient aggregation weights and performs multi-objective optimization to achieve fairness in federated learning. Cui et al. [17] used a smooth proxy for the maximum value function to consider the optimization objectives of all clients, thus making the global model more consistent across different clients. Hamer et al. [18] introduced a communication-efficient ensemble method called FedBoost for federated learning. This approach involves training an ensemble of pre-trained base predictors through federated learning. By solely learning the ensemble weights via federated learning, the algorithm enables the training of large models that may not fit within the memory constraints of client devices. Li et al. [19] achieved fairness by adjusting the client aggregation weights through a loss amplification mechanism and introducing a hyperparameter q to adjust the weights. Zhao et al. [20] improved global model fairness by dynamically adjusting the aggregation weights of client losses using a weight redistribution mechanism. Sun et al. [21] used a reinforcement learning algorithm to adaptively learn the aggregation weights for fairness, but this method significantly increased training costs. Li et al. [22] used the method of empirical risk minimization to set the aggregation weights to achieve a balance between model fairness and accuracy.

2.2. Based on Client-Sampling Strategies

It can determine the degree of participation of different clients in the training process and thus affect the degree of preference for the global model. Yang et al. [23] improved model fairness by allowing less frequently trained clients to participate more in training. Huang et al. [24] implemented long-term fairness constraints by setting a threshold such that the probability of each client not being selected for a long time is not less than the threshold, thus allowing each client to participate in training more fairly. Zhou et al. [25] improved fairness by selecting whether or not to train each client based on its network status, ensuring fairness under a certain packet loss rate. Wen et al. [26] proposed a client selection method called ChFL. ChFL assigns scores to clients based on their local training loss values and training time, where a higher score indicates a higher probability of a client being selected. However, ChFL still neglects the impact of data volume on the quality of local models. Hao et al. [27] improved fairness using zero-shot data augmentation for some clients to mitigate data heterogeneity. Fraboni et al. [28] improved model fairness using clustering and sampling to improve the representativeness of clients, but this method significantly increased training time and costs. Ghosh et al. [29] improved fairness by clustering clients into different classes and training a global model for each class, but this method goes against the original intention of federated learning for joint training and increases training costs.

2.3. Based on Personalized Local Models

This method trains local models simultaneously with the global model to improve model accuracy but also increases training costs. Li et al. [30] reduced the negative impact of data heterogeneity by making local updates approach global updates. Tian et al. [31] introduced FCFL, a performance optimization method that considers multiple data sources. FCFL assigns appropriate fair constraints, MCF, to each data source. Smoothing the surrogate maximum function minimizes the loss of the maximum local model and continues to optimize the model to Pareto optimality under the constraint of minimizing the maximum loss. However, the disadvantage of this method lies in its lack of robustness against malicious attacks. Li et al. [32] achieved a balance between fairness and robustness by adding a regularization term to train more accurate local models, but this significantly increases computational overheads.

3. Proposed Approach

In this section, we begin by presenting the relevant definitions pertaining to our proposed FedCSGP algorithm. Subsequently, we provide an overview of FedCSGP. Finally, we delve into a comprehensive and detailed explanation of FedCSGP.

3.1. Related Definitions

Gradient conflicts [33]: a conflict between client i and client j iff $G_i \cdot G_j < 0$, where G_i and G_j denote the gradients of client i and client j, respectively.

Gradient Projection [33]: It is a method that decomposes each gradient by computing the inner product between each pair of gradients, separating it into a component orthogonal to other gradients and a component related to other gradients. The calculation formula is as follows:

$$G_i^{\text{proj}} = G_i - \sum_{j \neq i} \left(\frac{G_i \cdot G_j}{\|G_j\|^2} \right) G_j \tag{1}$$

Internal conflicts [34]: it refers to the existence of gradient conflicts among the clients participating in the t-th communication round of training.

External conflicts [34]: it refers to the potential gradient conflicts between the participating clients i and the non-participating clients j during the t-th communication round of training.

3.2. Overview: FedCSGP Algorithm

In contrast to the FedAvg algorithm [35], the FedCSGP algorithm introduces a two-stage approach to the federated-learning training process: pre-training and formal training. The pre-training stage of the FedCSGP algorithm adopts a framework similar to that of the FedAvg algorithm, as depicted in Figure 1. Subsequently, the formal training stage follows a distinct framework, as illustrated in Figure 2.

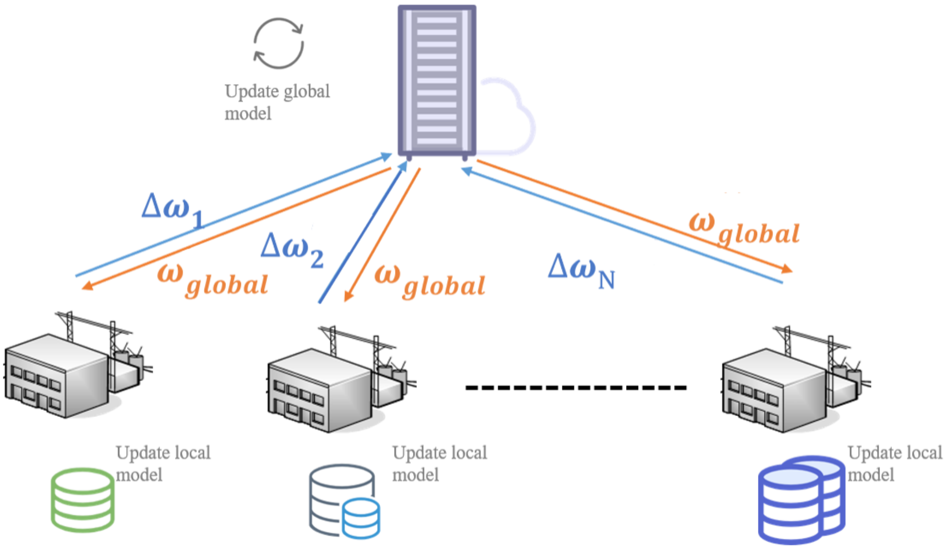


Figure 1. Framework of the pre-training phase of FedAvg.

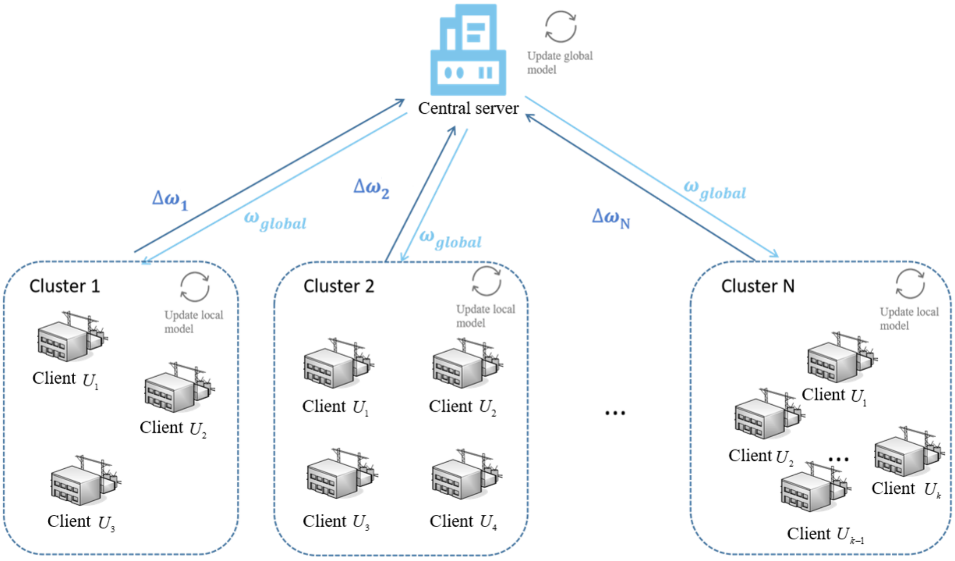


Figure 2. Framework of the formal training phase of FedCSGP.

The FedCSGP algorithm framework comprises a central server and a set of clients. During the pre-training phase, the central server organizes the electric power companies (referred to as clients) into multiple clusters. Once the formal training commences, the central server picks clients from each cluster for every training round and dispatches the global model to them. Using their respective local datasets, these chosen clients proceed with training and subsequently send their refreshed local models back to the central server. The central server processes and aggregates these local models to generate a new global model for the next round. The central server repeats this process by selecting clients from each cluster and distributing the updated global model. This iterative process continues until the training reaches the predefined number of rounds or the global model converges.

The FedCSGP algorithm aims to mitigate unfairness in federated learning by addressing external and internal conflicts. First, in the pre-training stage, a client-sampling strategy based on clustering is employed to alleviate external conflicts, therefore enhancing the representativeness and fairness of selected clients. This approach leads to reduced training rounds, lower communication costs, and faster model convergence. Second, in the formal training stage, the gradient-projection method is utilized to address internal conflicts by mitigating gradient discrepancies between clients with large and small training losses, enabling the attainment of a globally optimal model.

3.3. Detailed: FedCSGP Algorithm

The meanings of the notations employed in this paper are summarized in Abbreviations.

The detailed steps of the FedCSGP algorithm are as follows:

1. Pre-training phase

   Step 1: The central server initializes ω_0 and sends it to k clients.

   Step 2: Each client i trains the received global model ω_0 using its local data through gradient descent. After local iterative training, the client obtains a local model update $\Delta\omega_i$ and sends it to the central server, where $\Delta\omega_i = \omega_0 - \omega_{1_i}$.

   Step 3: The central server receives the uploaded local model updates $\Delta\omega_i$ from the clients and performs hierarchical clustering on these updates. This process generates C clusters and calculates the total data quantity NUM_j owned by each cluster.

   Step 4: The C clusters are sorted in descending order based on their total data quantities NUM_j. Then, the clusters are merged into m clusters according to the rule of having an equal data quantity in each cluster. The probability p_k of selecting each client within a cluster is computed.

2. Formal training phase

   Step 1: The central server randomly selects m clients from each cluster with probability p_k.

   Step 2: The central server sends the current global model ω_t to the selected clients, where ω_t represents the global model in the t-th round.

   Step 3: The selected clients train their local models ω_i^t using their respective local data through gradient descent. After local iterative training, each client obtains its local model ω_i^t and the corresponding loss value l_i^t in the t-th round.

   Step 4: The clients send their local models ω_i^t and loss values l_i^t to the central server.

   Step 5: The central server calculates the local model update $\mathrm{WD}_i^t = \omega_t - \omega_i^t$ for each client and stores them in W, i.e., $W = [\mathrm{WD}_1^t, \mathrm{WD}_2^t, \mathrm{WD}_3^t, \ldots, \mathrm{WD}_m^t]$.

   Step 6: The local model updates in W are sorted in ascending order based on their corresponding loss values. The sorted updates are stored in WO, i.e., $WO = [\mathrm{WD}_i^{t1}, \mathrm{WD}_i^{t2}, \mathrm{WD}_i^{t3}, \ldots, \mathrm{WD}_i^{tm}]$, where i represents the client index.

   Step 7: For each update $\mathrm{WD}_i^{t'}$ in WO, check if $\mathrm{WD}_i^{t'} \cdot \mathrm{WD}_i^t < 0$ holds, indicating an internal conflict between $\mathrm{WD}_i^{t'}$ and WD_i^t, where WD_i^t is from WO and $i \neq i'$. If $\mathrm{WD}_i^{t'} \cdot \mathrm{WD}_i^t < 0$ holds, it implies an internal conflict, and $\mathrm{WD}_i^{t'}$ needs to undergo orthogonal projection to mitigate the conflict. Compute $\mathrm{WD}_i^{t'} = \mathrm{WD}_i^{t'} - \frac{\mathrm{WD}_i^{t'} \cdot \mathrm{WD}_i^t}{|\mathrm{WD}_i^t|^2} \cdot \mathrm{WD}_i^t$.

   Step 8: Calculate the sum of the local model updates $\mathrm{WD}t = \frac{1}{m} \sum i = 1^m \mathrm{WD}_i^{t'}$.

   Step 9: Scale the sum of the projected local model updates WD_t by $\frac{|\mathrm{WD}_t|}{|\mathrm{WD}_t|}$.

Step 10: The central server updates the global model as $\omega_{t+1} = \omega_t - \mathrm{WD}_t$.

Repeat Steps 1 to 10 until the predetermined training rounds are reached.

The overall steps of the FedCSGP algorithm are shown in Algorithm 1.

Algorithm 1 FedCSGP Algorithm.

```
Input: $\omega_0$, $m$, $k$, $T$
Output: $\omega_t$
Server sends $\omega_0$ to k clients.
Client $i$ trains $\omega_0$ using its local dataset and obtains the local model update $\Delta\omega_i$. It then sends $\Delta\omega_i$ to the server. where $i = 1, 2, 3, \ldots, k$.
The server performs hierarchical clustering using $\Delta\omega_i$ as features, resulting in $C$ clusters and the data quantity $\mathrm{NUM}_j$ ,where $j = 1, 2, 3, \ldots, C$.
The $C$ clusters are sorted in descending order based on their data quantity $\mathrm{NUM}_j$, and then merged into $m$ clusters following the rule of equal data quantity per cluster.
for $t \in [1, 9]$ do
    The server randomly selects $m$ clients from the $m$ clusters with probability $p_k$.
    The server sends the global model $\omega_t$ to the selected clients.
    The selected clients train the received global model $\omega_t$ using their local data $D_i$.
    The clients obtain their local models $\omega_{ti}$ and the corresponding loss value $l_{ti}$ and upload them to the server.
    The server utilizes Algorithm 2, denoted as GPA$(\epsilon_t, \epsilon_{ti}, l_{ti})$, to output $\mathrm{WD}_i^{t'}$.
    The server calculates $\mathrm{WD}_t = \frac{1}{m}\sum_{i=1}^{m}\mathbf{W}\Delta_i^{t'}$.
    The server scales $\mathrm{WD}_t as \mathrm{WD}_t = \frac{|\mathbf{W}\Delta_t|}{|\mathbf{W}\Delta_t|}\mathrm{WD}_t$.
    The server updates the global model as $\omega_{t+1} = \omega_t + \mathrm{WD}_t$.
end for
```

Algorithm 2 Gradient project algorithm.

```
Input: $\omega_0$, $m$, $k$, $T$
Output: $\omega_t$
Server sends $\omega_0$ to k clients.
Client $i$ trains $\omega_0$ using its local dataset and obtains the local model update $\Delta\omega_i$. It then sends $\Delta\omega_i$ to the server. where $i = 1, 2, 3, \ldots, k$.
The server performs hierarchical clustering using $\Delta\omega_i$ as features, resulting in $C$ clusters and the data quantity $\mathrm{NUM}_j$ ,where $j = 1, 2, 3, \ldots, C$.
```

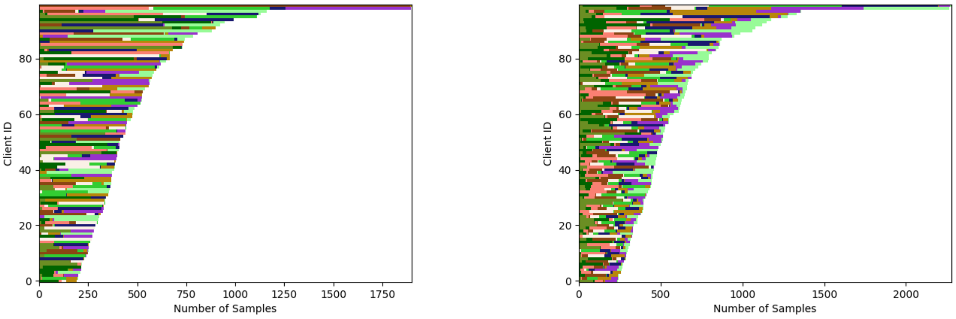
4. Experiments

4.1. Datasets

We evaluate the performance of FedCSGP on two publicly available datasets: MNIST [36] and CIFAR-10 [37]. To simulate a real-world scenario, we utilize the Dirichlet distribution [38] to partition these datasets into non-i.i.d. subsets with label distribution skew. This partitioning process considers the label distribution of the sample data, allocating different class labels to clients in disproportionate amounts. Once the number of clients is fixed, the final dataset distribution is shaped by two key parameters. The parameter b dictates the consistency of data volume across clients. Specifically, when $b = 0$, data volume is roughly equal among all clients, whereas when $b = 1$, there is a marked disparity in data volume between the clients with the most and least data. Additionally, the parameter α gauges the imbalance of the data distribution, ranging from 0.1 to 1. An increase in the value of α results in a more uneven distribution, heightening data heterogeneity.

For the MNIST dataset, with a total of 100 clients, we set $b = 0.5$ and $\alpha = 0.7$. For the CIFAR-10 dataset, with a total of 100 clients, we set $b = 0.6$ and $\alpha = 0.8$. The partitioned dataset distribution is visualized, where each row corresponds to a client's local dataset, and different colors represent different labels. The X-axis represents the size of the dataset collected by each client, and the Y-axis represents the client ID. The distribution of the

MNIST and CIFAR-10 datasets is shown in the respective Figure 3a,b. From the figures, it can be observed that CIFAR-10 exhibits higher data heterogeneity compared to MNIST. The variations in terms of the number of different labels, the quantity of samples, and the distribution among different clients are more pronounced in CIFAR-10.



(**a**) Data distribution of MNIST

(**b**) Data distribution of CIFAR-10

Figure 3. Data distribution of MNIST and CIFAR-10 in experiments. Visualize the distribution of the partitioned dataset, where each row corresponds to a client's local dataset, with different colors indicating various labels. The X-axis represents the quantity or size of the dataset collected by each client (Number of Samples), while the Y-axis denotes the client ID.

4.2. Experimental Setup

We selected the FedAvg algorithm as the baseline for federated learning. In addition, we chose the Clustered_Sampling algorithm [28] and the FedFV (federated fair averaging) algorithm [34] as the comparative approaches for addressing the fairness issues in federated learning. The experimental environment configuration parameters, including computer configuration parameters, development environment, and programming language used in this study, are presented in Table 1.

The other hyperparameters for federated learning are set as follows: the number of local iterations, denoted by E, is set to 5; the local batch size, denoted by B, is selected from the range [16, 32, 64]; the learning rate, denoted by η, is chosen from the range [0.01, 0.1]. To ensure fairness, the parameter settings for the four algorithms in the comparative experiments are kept consistent, and the other parameters are set to their respective optimal values, as mentioned in their respective papers.

Table 1. Experimental environment configuration parameters.

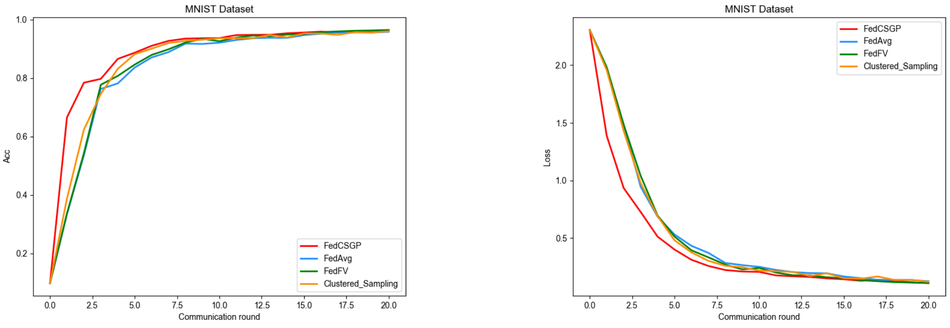
Parameter	Value
Operating System	Ubuntu 20.04
Memory	40 GB
CPU and GPU	Intel(R) Xeon(R) 8255C CPU, RTX 3080 (10 GB)
Development Framework	PyTorch 1.11.0
Programming Language	Python 3.8

4.3. Experimental Results

To comprehensively evaluate the fairness performance of the algorithms, accuracy trend graphs and loss value trend graphs are used to assess the accuracy and convergence speed of the algorithms on the test set. Experimental fairness is quantified using performance metrics related to fairness. The performance metric results are the average values obtained from five independent experiments. To ensure the reliability of the experimental results, each experiment is conducted with a different random seed.

4.3.1. MNIST

Experiments were conducted on the MNIST dataset using the FedCSGP, FedAvg, Clustered_Sampling, and FedFV algorithms. The accuracy trend graph and loss value trend graph are shown in Figure 4a,b. From the graphs, it can be observed that the final accuracy and loss values of these four algorithms converge to a similar level. However, the FedCSGP algorithm shows significantly faster improvement in accuracy and a faster rate of loss reduction compared to the other three algorithms. This indicates that the FedCSGP algorithm has an advantage in convergence speed and achieves a stable state ahead of the other algorithms. It can be inferred that the FedCSGP algorithm not only ensures accuracy but also exhibits faster convergence speed, which reduces the number of communication rounds and lowers communication costs.



(**a**) Data distribution of MNIST (**b**) Data distribution of CIFAR-10

Figure 4. The accuracy and loss of the four algorithms on the MNIST Dataset.

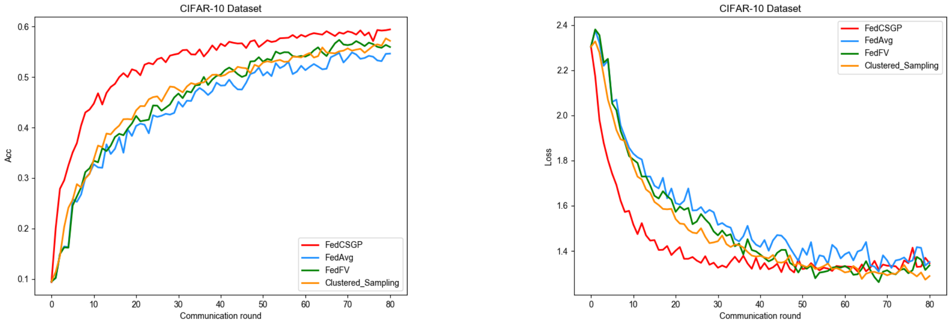
Table 2 presents the results of fairness-related performance metrics for the four algorithms on the MNIST dataset. The results show that the FedCSGP algorithm has the smallest value of Var(ω), with a reduction in variance of 28.37%, 10.17%, and 19.7% compared to the FedAvg, Clustered_Sampling, and FedFV algorithms, respectively. This indicates that among the four algorithms, the FedCSGP algorithm achieves the best fairness in the global model. Compared to the other three algorithms, the FedCSGP algorithm improves the lowest accuracy of the 100 clients by 2.6%, 2.2%, and 0.16%, respectively, without compromising the overall model accuracy. The accuracy of the worst 5% clients is also improved by 2.7%, 0.82%, and 1.3%, respectively. This demonstrates that during the training process, the FedCSGP algorithm promotes fairness in the global model for clients with poorer performance. Despite the fact that the FedCSGP algorithm necessitates clustering and gradient-projection operations to be carried out at the server, leading to a longer overall runtime compared to the FedAvg algorithm, the decrease in the number of communication rounds needed for FedCSGP to attain equilibrium, coupled with the significance of fairness considerations for clients, justifies this increase in runtime.

Table 2. Performance Metrics Comparison on MNIST Dataset.

Performance Metric	FedCSGP	FedAvg	FedFV	Clustered_Sampling
Var(ω)	0.00106	0.00148	0.00118	0.00132
Lowest Accuracy	0.83266	0.81094	0.81446	0.83132
Highest Accuracy	1	1	1	1
Average Accuracy	0.96062	0.95346	0.96216	0.95458
Top 5%	1	1	1	1
Worst 5%	0.87464	0.85104	0.8675	0.86366
Runtime	843.13628	442.20558	621.50996	1148.10524

4.3.2. CIFAR-10

The experimental validation was conducted on the FedCSGP algorithm and its three comparison algorithms. The experimental results show the accuracy and loss trends of the four algorithms on the test set, as depicted in Figure 5a,b. From Figure 5, it can be observed that the global model trained by the FedCSGP algorithm achieves significantly higher accuracy compared to the FedAvg, FedFV, and Clustered_Sampling algorithms. The FedCSGP algorithm also demonstrates faster convergence with a steeper increase in accuracy and a quicker decrease in loss, indicating its superior accuracy and convergence characteristics.



(**a**) Data distribution of MNIST

(**b**) Data distribution of CIFAR-10

Figure 5. The accuracy and loss of the four algorithms on the CIFAR-10 Dataset.

The fairness-related performance metrics of the four algorithms on the CIFAR-10 dataset are presented in Table 3. Although the FedCSGP algorithm has a variance 0.9% lower than the Clustered_Sampling algorithm, it outperforms the Clustered_Sampling algorithm in terms of the highest accuracy, average accuracy, and the accuracy of the worst 5% of clients by 7.9%, 2.5%, and 20.18%, respectively. Compared to the Clustered_Sampling algorithm, the FedCSGP algorithm achieves a 10.32% reduction in runtime. While the highest accuracy and top 5% accuracy of the FedCSGP algorithm are 1.9% lower than the FedFV algorithm among the 100 clients, its variance, lowest accuracy, average accuracy, and the accuracy of the worst 5% of clients are all significantly better than those of the FedFV algorithm. This indicates that the FedCSGP algorithm improves the fairness of the model while enhancing its overall accuracy. Therefore, the FedCSGP algorithm exhibits superior overall performance.

Throughout the training process, the FedCSGP algorithm significantly boosts the accuracy of underperforming clients without compromising or even enhancing the overall precision of the global model. This approach promotes fairness among all clients and facilitates quicker convergence, therefore cutting down on communication expenses.

Table 3. Performance metrics on the CIFAR-10 dataset.

Performance Metric	FedCSGP	FedAvg	FedFV	Clustered_Sampling
Var(ω)	0.01532	0.01796	0.01944	0.01518
Lowest Accuracy	0.31622	0.23714	0.24056	0.29294
Highest Accuracy	0.92018	0.90702	0.93846	0.91292
Average Accuracy	0.59628	0.5628	0.58618	0.58192
Best 5% Accuracy	0.85388	0.85126	0.875	0.8411
Worst 5% Accuracy	0.35244	0.30652	0.33504	0.33226
Runtime (seconds)	3474.92874	1963.22436	2511.23318	3874.9994

Furthermore, during the experimentation process, it was observed that FedAvg and Clustered_Sampling algorithms had certain requirements for the learning rate. Specifically, these algorithms were prone to gradient explosion during training, leading to training failure. In comparison, FedCSGP and FedFV algorithms exhibited higher tolerance towards the learning rate and were less susceptible to gradient explosion.

In summary, on the MNIST and CIFAR-10 datasets, compared to the algorithms chosen in this paper: FedAvg, FedFV, and Clustered_Sampling, the FedCSGP algorithm demonstrates exceptional performance and advantages in terms of accuracy, loss value, convergence speed, and stability. The FedCSGP algorithm enhances the accuracy of poorly performing clients while ensuring the accuracy of the global model, therefore improving the fairness of the federated-learning approach. In addition, the FedCSGP algorithm reduces the number of communication rounds, therefore lowering communication overhead. Based on this, the proposed algorithm is suitable for application in smart-grid scenarios.

5. Conclusions and Future Work

In this paper, we propose the FedCSGP algorithm for the smart grid, which divides the model-training process into federated learning into two stages: pre-training and formal training. In the pre-training stage, external conflicts are mitigated through a client-sampling strategy based on clustering, while in the formal training stage, internal conflicts are alleviated using the gradient-projection algorithm, therefore improving the fairness of the model. Additionally, experimental results on the MNIST and CIFAR-10 datasets demonstrate that FedCSGP achieves enhanced fairness and reduced communication costs compared to the baseline algorithm while ensuring accuracy. In our future endeavors, we intend to concentrate on bolstering privacy and security within federated learning, aiming to achieve a harmonious equilibrium between accuracy, fairness, and privacy.

Author Contributions: Conceptualization, R.Z. and Z.L.; methodology, C.H.; software, R.Z.; validation, J.L. and T.W.; formal analysis, W.G.; investigation, Z.L. and T.L.; resources, C.H.; data curation, J.L.; writing—original draft preparation, T.L.; writing—review and editing, R.Z.; visualization, C.H.; supervision, T.L.; project administration, Z.L.; funding acquisition, W.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Science and Technology Project of China Southern Power Grid Corporation (No. 03600OKK52222018) (No. GDKJXM20222155).

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: Author Ruifeng Zhao, Jiangang Lu, Wenxin Guo and Tian Lan were employed by the company Electric Power Dispatching and Control Center of Guangdong Power Grid Co. Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

Notation	Meaning
ω_0	Initialized global model
k	Total number of clients
T	Communication rounds
$\Delta\omega_i$	Local model update of client *i* in the pre-training phase
ω_1^i	Local model trained by client *i* in the pre-training phase
C	Number of classes obtained in the pre-training phase
NUM_j	Total data quantity of all clients in class *j*
m	Number of clients sampled per training round
p_k	Probability of client *k* being selected

ω_t	Global model in the t-th round
ω_i^t, l_i^t	Local model and loss value of client i in the t-th round
WD_i^t	Local model update of client i in the t-th round
W	List of local model updates of clients before sorting
WO	List of local model updates of clients after sorting
$\mathrm{WD}_i^{(t')}$	Projected size of local model update of client i
WD^t	Sum of projected local model updates

References

1. Taghia, J.; Moradi, F.; Larsson, H.; Lan, X.; Orucu, A.; Ebrahimi, M.; Johnsson, A. Congruent learning for self-regulated federated learning in 6g. *IEEE Trans. Mach. Learn. Commun. Netw.* **2024**, *2*, 129–149. [CrossRef]
2. Voigt, P.; Von dem Bussche, A. The eu general data protection regulation (gdpr). In *A Practical Guide*, 1st ed.; Springer International Publishing: Cham, Switzerland, 2017; Volume 10, pp. 10–5555.
3. de la Torre, L. A Guide to the California Consumer Privacy Act of 2018. *SSRN Electron. J.* **2018**, 1–8. SSRN 3275571.
4. Han, S.W.; Munir, A.B. Information security technology-personal information security specification: China's version of the gdpr. *Eur. Data Prot. Law Rev.* **2018**, *4*, 535.
5. Liu, X.; Deng, Y.; Nallanathan, A.; Bennis, M. Federated learning and meta learning: Approaches, applications, and directions. *IEEE Commun. Surv. Tutor.* **2024**, *26*, 571–618. [CrossRef]
6. Kalapaaking, A.P.; Khalil, I.; Rahman, M.S.; Atiquzzaman, M.; Yi, X.; Almashor, M. Blockchain-based federated learning with secure aggregation in trusted execution environment for internet-of-things. *IEEE Trans. Ind. Inform.* **2023**, *19*, 1703–1714. [CrossRef]
7. He, S.; Shi, K.; Liu, C.; Guo, B.; Chen, J.; Shi, Z. Collaborative sensing in internet of things: A comprehensive survey. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 1435–1474. [CrossRef]
8. Dressel, J.; Farid, H. The accuracy, fairness, and limits of predicting recidivism. *Sci. Adv.* **2018**, *4*, eaao5580. [CrossRef] [PubMed]
9. Du, W.; Xu, D.; Wu, X.; Tong, H. Fairness-aware agnostic federated learning. In Proceedings of the 2021 SIAM International Conference on Data Mining (SDM), Virtual Event, 29 April–1 May 2021; SIAM: Philadelphia, PA, USA, 2021; pp. 181–189.
10. Briggs, C.; Fan, Z.; Andras, P. Federated learning with hierarchical clustering of local updates to improve training on non-iid data. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–9.
11. Ghosh, A.; Hong, J.; Yin, D.; Ramchandran, K. Robust federated learning in a heterogeneous environment. *arXiv* **2019**, arXiv:1906.06629.
12. Smith, V.; Chiang, C.-K.; Sanjabi, M.; Talwalkar, A.S. Federated multi-task learning. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 4427–4437.
13. Kairouz, P.; McMahan, H.B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A.N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. Advances and open problems in federated learning. *Found. Trends® Mach. Learn.* **2021**, *14*, 1–210. [CrossRef]
14. Mohri, M.; Sivek, G.; Suresh, A.T. Agnostic federated learning. In *International Conference on Machine Learning*; PMLR: New York, NY, USA, 2019; pp. 4615–4625.
15. Wang, J.; Liu, Q.; Liang, H.; Joshi, G.; Poor, H.V. Tackling the objective inconsistency problem in heterogeneous federated optimization. *arXiv* **2020**, arXiv:2007.07481.
16. Hu, Z.; Shaloudegi, K.; Zhang, G.; Yu, Y. Federated learning meets multi-objective optimization. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 2039–2051. [CrossRef]
17. Cui, S.; Pan, W.; Liang, J.; Zhang, C.; Wang, F. Addressing algorithmic disparity and performance inconsistency in federated learning. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 26091–26102.
18. Hamer, J.; Mohri, M.; Suresh, A. Fedboost: A communication-efficient algorithm for federated learning. *PMLR* **2020**, *119*, 3973–3983.
19. Li, T.; Sanjabi, M.; Beirami, A.; Smith, V. Fair resource allocation in federated learning. *arXiv* **2019**, arXiv:1905.10497.
20. Zhao, Z.; Joshi, G. A dynamic reweighting strategy for fair federated learning. In Proceedings of the ICASSP 2022—2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Virtual, 7–13 May 2022; pp. 8772–8776.
21. Sun, Y.; Si, S.; Wang, J.; Dong, Y.; Zhu, Z.; Xiao, J. A fair federated learning framework with reinforcement learning. In Proceedings of the 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy, 18–23 July 2022; pp. 1–8.
22. Li, T.; Beirami, A.; Sanjabi, M.; Smith, V. Tilted empirical risk minimization. *arXiv* **2020**, arXiv:2007.01162.
23. Yang, M.; Wang, X.; Zhu, H.; Wang, H.; Qian, H. Federated learning with class imbalance reduction. In Proceedings of the 2021 29th European Signal Processing Conference (EUSIPCO), Dublin, Ireland, 23–27 August 2021; pp. 2174–2178.
24. Huang, T.; Lin, W.; Wu, W.; He, L.; Li, K.; Zomaya, A.Y. An efficiency-boosting client selection scheme for federated learning with fairness guarantee. *IEEE Trans. Parallel Distrib. Syst.* **2020**, *32*, 1552–1564. [CrossRef]
25. Zhou, P.; Fang, P.; Hui, P. Loss tolerant federated learning. *arXiv* **2021**, arXiv:2105.03591.
26. YiLin, W.; Zhao, N.; Yan, Z. Client selection method based on local model quality. *Comput. Eng.* **2023**, *49*, 131–143.
27. Hao, W.; El-Khamy, M.; Lee, J.; Zhang, J.; Liang, K.J.; Chen, C.; Duke, L.C. Towards fair federated learning with zero-shot data augmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 3310–3319.

28. Fraboni, Y.; Vidal, R.; Kameni, L.; Lorenzi, M. Clustered sampling: Low-variance and improved representativity for clients selection in federated learning. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021; pp. 3407–3416.
29. Ghosh, A.; Chung, J.; Yin, D.; Ramchandran, K. An efficient framework for clustered federated learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 19 586–19 597. [CrossRef]
30. Li, T.; Sahu, A.K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; Smith, V. Federated optimization in heterogeneous networks. *Proc. Mach. Learn. Syst.* **2020**, *2*, 429–450.
31. Jiahui, T.; Lv, T.; Zhou, R. A fair resource allocation scheme in federated learning. *J. Comput. Res. Dev.* **2022**, *59*, 1240–1254.
32. Li, T.; Hu, S.; Beirami, A.; Smith, V. Ditto: Fair and robust federated learning through personalization. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021; pp. 6357–6368.
33. Yu, T.; Kumar, S.; Gupta, A.; Levine, S.; Hausman, K.; Finn, C. Gradient surgery for multi-task learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 5824–5836.
34. Wang, Z.; Fan, X.; Qi, J.; Wen, C.; Wang, C.; Yu, R. Federated learning with fair averaging. *arXiv* **2021**, arXiv:2104.14937.
35. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.
36. LeCun, Y.; Cortes, C.; Burges, C. Mnist Handwritten Digit Database. 2010. Available online: http://yann.lecun.com/exdb/mnist/ (accessed on 19 May 2024).
37. Krizhevsky, A.; Hinton, G. Learning Multiple Layers of Features from Tiny Images. 2009. Available online: https://www.cs.toronto.edu/~kriz/cifar.html (accessed on 19 May 2024).
38. Hsu, T.-M.H.; Qi, H.; Brown, M. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv* **2019**, arXiv:1909.06335.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.