Personalized Federated Learning with Adaptive Feature Extraction and Category Prediction in non-IID Datasets

Ying Hsun Lai , Shin Yeh Chen , Wen Chi Chou , Hua-Yang Hsu * , Han-Chieh Chao *

Posted Date: 24 January 2024

doi: 10.20944/preprints202401.1769.v1

Keywords: personalized federated learning; client drift; non-IID Datasets; neural network

Article

Personalized Federated Learning with Adaptive Feature Extraction and Category Prediction in Non-IID Datasets

Ying-Hsun Lai [1], Shin-Yeh Chen [2], Wen-Chi Chou [3], Hua-Yang Hsu [4,*], and Han-Chieh Chao [5,6*]

[1] Department of Computer Science and Information Engineering, National Taitung University, Taitung, Taiwan; yhlai@nttu.edu.tw
[2] Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung, Taiwan; sychen@mail.ntou.edu.tw
[3] Taiwan Semiconductor Manufacturing Company, Hsinchu, Taiwan; wcchou2001@tsmc.com
[4] School of Electronic and Computer Engineering (SECE), Peking University Shenzhen Graduate School, Shenzhen, China
[5] Department of Electrical Engineering, National Dong Hwa University, Hualien, Taiwan
[6] Computer Science and Innovation, UCSI University, Kuala Lumpur, Malaysia
* Correspondence: huayang.hsu@gmail.com (H.-Y.H.); hcc@mail.ndhu.edu.tw (H.-C.C.); Tel.: +886-3-890-6006

Abstract: Federated learning trains a neural network model using the client's data to maintain the benefits of centralized model training, while maintaining the privacy. However, if the client data are not independently and identically distributed (non-IID) because of different environments, the accuracy of the model may suffer from client drift during training owing to discrepancies in each client's data. This study proposes a personalized federated learning algorithm based on the concept of multitask learning to divide each client model into two layers: a feature extraction layer and a category prediction layer. The feature extraction layer maps the input data to a low-dimensional feature vector space. Furthermore, the parameters of the neural network are aggregated with those of other clients using an adaptive method. The category prediction layer maps low-dimensional feature vectors to the label sample space, with its parameters remaining unaffected by other clients to maintain client uniqueness. The proposed personalized federated learning method produces faster learning model convergence rates and higher accuracy rates for the non-IID datasets in our experiments.

Keywords: personalized federated learning; client drift; non-IID Datasets; neural network

1. Introduction

Artificial intelligence (AI) has rapidly developed with the continuous advancement of technology. Deep learning is one of the most popular AI technologies. The main feature of deep learning technology is that it can automatically extract features from input data and train efficient neural network models using large amounts of data [1–3]. However, deep learning technology requires a large amount of data for training, and these data usually come from different clients. The challenge of effectively training deep learning models while preserving client data privacy has emerged as a critical issue. Therefore, customer data privacy has become a major concern in deep learning technology [4,5].

Federated learning technology has emerged as a solution for avoiding problem of exposing client data during model training. It is a distributed machine learning method that can aggregate data for model training while maintaining client privacy [6,7]. The main concept of federated learning technology is to divide the model training process into multiple stages, each of which involves different clients and transmits only the model parameters. This approach safeguards the confidentiality of client data when using large datasets for model training, resulting in enhanced model accuracy and generalization. Federated learning technology has found extensive application in various domains, including healthcare, finance, and intelligent transportation [8–10].

 doi:10.20944/preprints202401.1769.v1

Despite significant progress in federated learning, some technical challenges remain unaddressed. One of the main issues is non-independent and identically distributed (non-IID) data heterogeneity. In real-world applications, data distribution among clients often varies. In federated learning, every device or server has its own data distribution, which can be significantly different from that of other devices. This can lead to poor performance if the model is trained using non-IID data [11,12]. This study presents a novel approach to federated learning that allows each client to have its own personalized model and preserves local training results during the federated learning process. This approach effectively mitigates the impact of non-IID data on federated learning. To achieve this, the neural network model is divided into two parts: a feature extraction layer and a category prediction layer. During the federated learning process, each client has its own personalized model, which follows the above two-part structure. When the server returns the new global model to the clients, the category prediction layer of each client is unaffected and retains its local training results. The feature extraction layer does not change completely according to the global model. Instead, it is compared with the global model and local training results and then mixed based on the comparison results. Using this approach, this study aims to effectively mitigate the impact of non-IID data on federated learning and improve its performance. The contributions of this study can be summarized as follows:

1. This study aims to develop a personalized federated learning process to mitigate the impact of data heterogeneity on federated learning.

Unlike the conventional federated learning method, which trains a global model for all clients, this study proposes a personalized approach that adjusts the global model based on the data distribution of each client.

2. A each client.

This study proposes a theoretical basis for client model partitioning based on the properties of neural network models and presents a practical partitioning method that divides client models into feature extraction and category prediction layers. During federated learning training, the feature extraction layers are adjusted in a personalized manner, whereas the category prediction layers serve as elements that give the client models personalized characteristics.

3. This method includes personalized feature extraction for client data.

Federated learning requires dataset with similar properties to achieve results similar to those of multi-task learning by finding an excellent feature extraction function. However, the effect of client drift makes it difficult for the feature extraction layer to achieve excellent feature extraction capabilities, which affects the final performance. Therefore, this study proposes an adaptive approach to reduce the adverse effects of client drift on the feature extraction layer.

2. Related Work

4. neural network model is used to establish personalized models for The concept of federated learning was first presented by Google in 2017. This facilitates the training of machine learning models without centralized data. Instead, user data are stored on the client's side, and all clients participate in the training process. The primary objective of federated learning is to safeguard user privacy and achieve a more generalized model. During training, client-side processing exclusively handles user data, and the model gradient undergoes encryption upon returning to the server to prevent access by other clients. Furthermore, in federated learning, the server-side aggregation algorithm considers all client neural network model parameters to yield a more generalized model. McMahan et al. proposed a Federated Averaging (FedAvg) algorithm to aggregate client model parameters on the server side in federated learning [13]. The FedAvg algorithm computes the average of the client model parameters and employs them as a global model for a specific round.

The main distinction between federated and traditional deep learning methods is the dataset environment. In traditional deep learning, the dataset is centralized to provide model with access to all data. However, in federated learning, clients independently own their datasets, resulting in various data distributions and impacting the model performance. Based on the state of the client dataset, federated learning can be categorized into independent and identically distributed (IID) or

 doi:10.20944/preprints202401.1769.v1

non-IID. Non-IID datasets reduce the effectiveness of federated learning owing to client drift, as explained by Karimireddy et al. [14]. When the client datasets are non-IIDs, the local models of each client get updated in different directions, resulting in conflicting effects during aggregation on the server and decreasing the effectiveness of federated learning. Karimireddy et al. (2020) also suggested that client data heterogeneity could cause client drift, reducing the efficiency and accuracy of federated learning [15]. Wang et al. utilized reinforcement learning to minimize the effects of non-IID datasets on a global model and increase the convergence rate of federated learning [16]. Zhao et al. investigated and analyzed the performance of the FedAvg algorithm in an environment where the client was distributed with non-IID datasets [17]. The study found that the severity of non-IID datasets significantly affected the performance of the models trained using FedAvg. Li et al. proposed the FedProx algorithm to enhance the vulnerability of the FedAvg algorithm to heterogeneous client data by modifying the training objective function of the client model [18].

Personalized federated learning is a novel machine learning technique that addresses the issue of data heterogeneity in federated learning. In traditional federated learning, a global model is shared among all clients; however, when the data are non-uniformly distributed among clients, the efficacy of the approach is compromised. Personalized federated learning creates an individualized model for each client that adapts to the heterogeneity of their datasets, thereby enhancing the effectiveness of the federated learning. Wu et al. examined the limitations of traditional federated learning when dealing with various forms of heterogeneity, and suggested that personalized federated learning can effectively resolve the heterogeneity issue [19]. FedTP developed a learn-to-personalized mechanism to encourage cooperation among clients and to increase scalability and generalization. The results show that FedTP performed better than FedAvg-T in all cases. FedTP can combined the FedPer, FedRod, and KNN-Per methods to enhance model performance [30]. Li et al. (2021) presented the Ditto algorithm, which is comparable to FedProx, by incorporating a proximal term into the client's objective function to indicate the divergence between the client's local model and the global model [20]. Moreover, they regulate the extent of the individualization of the client's local model. Arivazhagan et al. proposed FedPer as a personalized federated learning method to address problems associated with client data heterogeneity [21]. FedPer divides the model into base and personalized layers. The base layer was jointly trained on all client data, whereas the personalized layer was trained solely on the client's local data. The experimental findings indicate that FedPer outperforms the general federated learning method, FedAvg, when dealing with heterogeneous client data. However, FedPer lacks a clear definition of how to partition personalized models, and the base layer is obtained through FedAvg, which results in the base layer being affected by client data heterogeneity. To address this, our study suggests partitioning the model into feature extraction and category prediction layers, while obtaining the feature extraction layer through an adaptive method rather than FedAvg. Additionally, this study introduces the concept of layer partitioning and separates the model into a feature extraction layer and a category prediction layer based on the distinct characteristics of the neural network layers. Review of the related work is shown in Table 1 below.

Table 1. Related work review.

Method	Characteristic	Research Target & Advantages
FedAvg [13]	The first federated learning algorithm proposed by Google.	FedAvg is a collaborative training neural network with data privacy.
Mime [15]	It combines control-variates and server-level optimizer state.	Mime overcomes the natural client-heterogeneity and is faster than any centralized method.

 doi:10.20944/preprints202401.1769.v1

FAVOR [16]	It proposes a new method based on Q-learning to select a subset of devices.	FAVOR focuses on validation accuracy and penalizes the use of more communication rounds.
FedProx [18]	FedProx allows for the emergence of inadequately trained local models and adds proximal term to the clients' loss function	FedProx reduces the impact of non-IID on the federated learning and improves the accuracy relative to FedAvg.
FedPer [21]	It separates the client model into two parts and trains individually	FedPer demonstrates the ineffectiveness of FedAvg and the effectiveness in modeling personalization tasks.
FedTP [30]	It learns personalized self-attention for each client while aggregating the other parameters among the clients.	It combines FedTP with the other methods, including FedPer, FedRod and KNN-Per, to further enhance the model performance. It achieves better accuracy and learning performance.
Proposed Method	It proposes a personalized federated learning with adaptive feature extraction and category prediction.	The study shows faster convergence speed and lower data loss than the FedProx and the FedPer federated learning algorithms in Fashion-MNIST, CIFAR10, and CIFAR100 datasets.

3. Research Methodology

4.1. Personalized Client Models

This study aims to achieve personalized federated learning and practical division methods based on the characteristics of client neural network models. In this study, all client neural network models are tasked with image recognition, and the data dimensions of the input space R^x and prediction output space R^y are equal. Therefore, each client's image recognition task can be divided into feature extraction and category prediction tasks. The objective of feature extraction is to map the input data residing in a high-dimensional feature space, to a lower-dimensional feature space. In this process, it is necessary to maintain the separability of the input data while achieving dimensionality reduction. In this study, neural networks are used for feature extraction, and the corresponding function is defined as $F: R^x \rightarrow R^l$, where R^x is the data input space and R^l is the low-dimensional feature space. Similarly, neural networks will be used for category prediction in this study, and the corresponding function is defined as $G: R^l \rightarrow R^y$, where R^y is the category prediction output space. The image recognition task of each client neural network model can be represented as Equation (1), where the input data $x_i \in R^x$ and the model output $y_i \in R^y$.

$$yi = softmax\,(G\,(F\,(xi))) \qquad (1)$$

In the IID dataset, the data distribution of each client is the same as the global data distribution; therefore, feature extraction F and category prediction G are not affected by client drift. However, individual client data distributions are not representative of the global data distribution, and the direct aggregation of client models can severely affect the learning performance of the neural network owing to client drift. With multitask learning technology, a neural network model can achieve good

 doi:10.20944/preprints202401.1769.v1

results and improve the generalization of the model input by simultaneously learning multiple related tasks [22]. This study integrates the principles of federated learning with multi-task learning in a non-IID dataset. As the data distribution of each client cannot represent the global data distribution, each client is regarded as a different task with relatedness in multi-task learning. The effectiveness of multitask learning led to the discovery of a global feature extraction neural network. Consequently, each client possesses a feature extraction function for obtaining effective features from the input data. The client neural network model comprises a Convolutional Neural Network (CNN) and a Fully Connected Neural Network (FCNN), as shown in Figure 1.

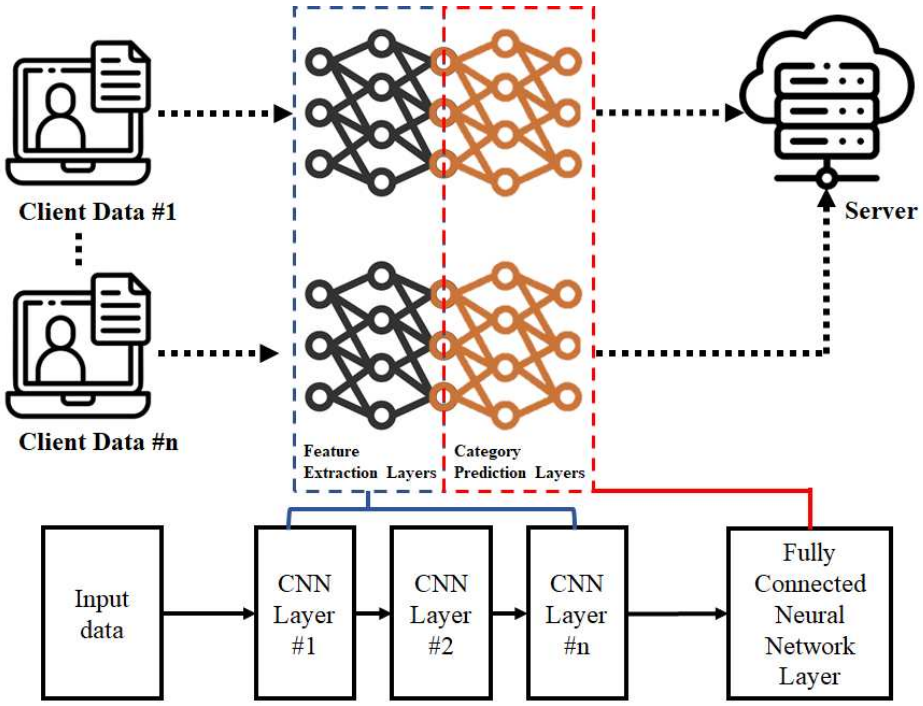


Figure 1. Personalized Client Neural Network Model.

The CNN kernel function considers the relationships between a single pixel and a hidden layer neuron and between the neighboring pixels. This renders CNNs efficient and resilient in extracting image features. In this study, the CNN of each client model serves as a feature extraction method to obtain a global feature function. However, in a FCNN, all the dimensions of the input data are connected to the hidden layer neurons, and the distance between the input features is not considered. This property effectively considers all the features and maps them to the prediction output space to achieve the function of a classifier. The FCNN part of the client model serves as the category prediction layer and is trained separately based on the data distribution of each client to achieve personalized properties.

4.1. Adaptive Feature Extraction

The learning in neural network models is significantly affected by both the initial setting values of the hyperparameters and the lack of corresponding changes in these values with increased training times. The adaptive learning method involves analyzing the dataset output results using a neural network model throughout the training process [23]. This leads to automatic adjustment of the hyperparameters, learning rates, and sometimes even the parameters of the neural network model, based on the analysis. The adaptive learning approach in this study adapts the APFL method [24]. Each client's feature extraction layer is obtained through a blend of global and client model parameters from the previous round, with the blending ratio established via adaptation. F_i^w denotes the parameter of the feature extraction layer, while G_i^w denotes the parameter of the category prediction layer, where i ∈ {1, 2...n} represents the i-th client, and * represents the best solution of the

 doi:10.20944/preprints202401.1769.v1

model, and D_i is used as the distribution of each client dataset. It can also represent the neural network model trained based on the data distribution D as $(F^w \circ G^w) \sim D$. In the IID dataset, the federated learning relationship between each client model and the global model can be represented using Equation (2).

$$((F_1^w \circ G_1^w) \sim D_1 \approx ((F_2^w \circ G_2^w) \sim D_2 \approx \cdots \approx ((F_n^w \circ G_n^w) \sim D_n$$

$$due\ to\ D_1 = D_2 = \cdots = D_n$$

$$(F_*^w \circ G_*^w) \approx \frac{1}{n}\sum_{i=1}^{n}((F_i^w \circ G_i^w) \quad (2)$$

In Equation (2), because the data distribution is IID, each client's data distribution can also represent the global data distribution. Therefore, in addition to the similarity between the model parameters for each client, they are closer to the best solution of the global model in that round. Therefore, it is less affected by the client drift problem and can provide satisfactory training results. However, when the data distribution is non-IID, the model parameters of each client are determined by different data distribution of each client, resulting in different model update directions. Therefore, for non-IID data distribution in federated learning, the relationship between each client model and the global model can be represented using Equation (3).

$$((F_1^w \circ G_1^w) \sim D_1 \not\approx ((F_2^w \circ G_2^w) \sim D_2 \not\approx \cdots \not\approx ((F_n^w \circ G_n^w) \sim D_n$$

$$due\ to\ D_1 \neq D_2 \neq \cdots \neq D_n$$

$$(F_*^w \circ G_*^w) \not\approx \frac{1}{n}\sum_{i=1}^{n}((F_i^w \circ G_i^w) \quad (3)$$

When a server aggregates the client models, they compete with each other. Even if a few client models are closer to the global best model update direction, they will still be affected by the incorrect update direction of other client models. This results in a severe client drift and poor performance of the aggregated global model. When each client data set is non-IID, a good global output model can only be obtained by aggregating a few client models that are closer to the global best model update direction. In a personalized federated learning method, each client obtains a personalized model M_k^w as shown in Equation (4).

$$M_k^w = (\frac{1}{n}\sum_{i=1}^{n}(F_i^w \sim D_i)) \circ (G_n^w \sim D_n) \qquad k \in \{1, 2 \ldots n\} \quad (4)$$

The feature extraction layer of M_k^w is obtained using the general federated learning aggregation method. Therefore, this study uses a mixture of global and previous round client model parameters to obtain feature extraction and reduce the impact of client drift, as shown in Equation (5).

$$M_k^{w\prime} = (((1-\beta) * \frac{1}{n}\sum_{i=1}^{n}(F_i^w \sim D_i) + (\beta * F_n^w \sim D_n))) \circ (G_n^w \sim D_n) \quad (5)$$

The mixing ratio between the global model parameters and the previous round client model parameters is determined by the β value ($0 \le \beta \le 1$, $\beta \in R$). When $\beta \to 0$, it represents that the mixed feature extraction terms will be dominated by the global model parameters. Conversely, when $\beta \to 1$, it represents that the mixed feature extraction terms will be dominated by the previous round client model parameters. Therefore, the β needs to be adjusted based on the gradient before each round of client learning to achieve an appropriate mixing ratio. Using this adaptive method, the impact of client drift on the feature extraction layer can be reduced.

In this study, the total time complexity includes a convolutional layer for feature extraction, a fully connected layer for category prediction and personalized federated learning. The time complexity of convolutional layer is expresseds as $O(n \times m)$, the time complexity of fully connected layer is $O(n \times m)$ and the time complexity of the personal federated learning model training is $O(w \times e \times n \times m)$, where w is the client number, e is the number of model iterations, n is the number of datasets, m is the number of parameters [31].

 doi:10.20944/preprints202401.1769.v1

The following algorithm is based on the division and separate training of each client model, and then incorporates the adaptive mixing method of feature extraction layer parameters.

Algorithm *adaptive algorithm in federated learning model*

let W_f be the parameters of feature extraction layers .

let W_g be the parameters of category predictions layer.

let i be i_{th} global round.

let c be the c_{th} selected client.

let γ be ths learning rate

let β be ths mixing ration, and the initial value is 0.5.

for i=1,2...,N **do**

setβ to initial value

if i==N **then** all clients do

new β = *betaupdate* (β)

client own model $\leftarrow ((1-\beta) * W_f^{i-1} + \beta * W_f^{i-1,c}) \circ W_g^{i-1,c}$)

break

else each selected clients $C \in \{C_1, C_2 ... C_m\}$ parallel do

receive W_f^{i-1} from server.

new β = *betaupdate* (β)

$(W_f^{i,c} \circ W_g^{i,c}) \leftarrow$ *modelupdate* $(((1-\beta) * W_f^{i-1} + \beta * W_f^{i-1,c}) \circ W_g^{i-1,c}, \gamma)$

keep the $W_g^{i,c}$

send $W_f^{i,c}$ back to server

finish the federated learning

4.1. Adaptive Mixing Ratio - β

In the local model training process of each client in the federated learning process, it is necessary to update the β value through gradient descent to obtain a suitable mixing ratio. In general image recognition learning, the loss value L can be expressed using Equation (6), where M is the neural network model, I is the input data, D_{label} is the labeled data and f_{loss} is the cross-entropy loss function.

$$L = f_{loss}(M(I), D_{label}) \tag{6}$$

After calculating the loss between the prediction results and the true label data using Equation (6), the neural network model must update its parameters based on the loss through gradient descent to reduce the gap. The loss value of each client can be extended based on the loss function of the general image recognition learning task, as shown in Equation (7), where F_g is the global feature extraction term, F_l^i is the local feature extraction term of the i-th client, and G_l^i is the gradient descent of the i-th client.

$$L_i = f_{loss}(((1-\beta_i) * F_g + \beta_i * F_l^i) \circ G_l^i(I), D_{label}) \tag{7}$$

In Equation (7), each client has its own loss value due to different model parameters, so each client can update β_i value according to the corresponding loss through gradient descent. And the model feature extraction term will be calculated with the (1-β_i) global model parameters and the β_i client model parameters of previous round. In addition to updating the β_i value through gradient descent to adjust the mixing ratio, the initial β value will also impact the client's accuracy in that round. The results of personalized federated learning using various initial β values of 0.3, 0.5, and 0.7

 doi:10.20944/preprints202401.1769.v1

are presented in Figure 2. According to the training results, when the initial value of β is closer to 1, such as 0.7, the training results are better than the other two values, which is consistent with Equation (6).

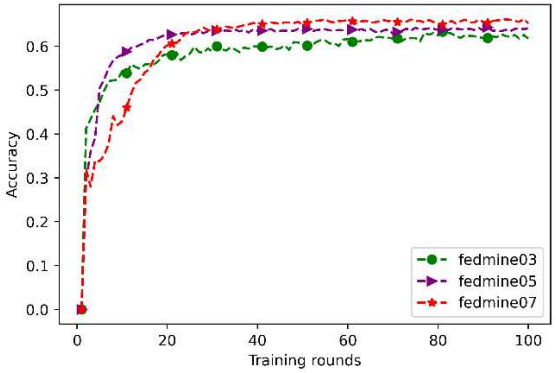


Figure 2. Learning results for different initial β values (β = 0.3, 0.5, 0.7).

4. Experiment Results and Analysis

4.1. Experimental Framework and Dataset

In this study, the FLOWER federated learning framework was used as the implementation framework for the proposed federated learning algorithm as shown in Figure 3. The FLOWER framework, developed by the German company Adap, has high scalability and supports several machine learning frameworks, including Pytorch, TensorFlow/Keras, and scikit-learn [25]. The framework is designed based on practical scenarios, enabling it to manage a large number of client loads and operate with heterogeneous client operating systems. Consequently, numerous researchers have used the FLOWER framework extensively as a federated learning implementation framework [26,27].

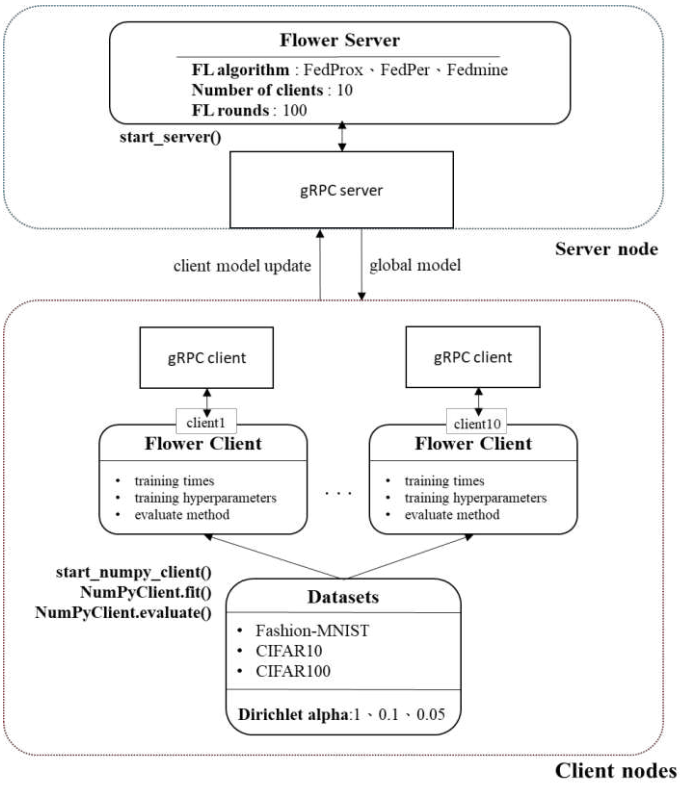


Figure 3. Federated learning with FLOWER framework.

 doi:10.20944/preprints202401.1769.v1

In this study, the image recognition task is chosen as the training target for the federated learning neural network model, and three image datasets including Fashion-MNIST [28], CIFAR-10 [29], and CIFAR-100 [29] are used to train the neural network model. These three datasets are commonly used for training neural network models for deep learning. The Fashion-MNIST dataset mainly contains popular fashion items, including T-shirts/tops, pants, pullovers, dresses, coats, sandals, shirts, sneakers, bags, and boots, where image is grayscale and has a size of 28 pixels. The CIFAR-10 dataset contains 10 image categories including airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks, and each image has a size of 32 pixels. However, the primary difference between the CIFAR-10 and Fashion-MNIST datasets is that each image in the CIFAR-10 dataset has three color channels, which makes the training of the neural network model more difficult. The CIFAR-10 and CIFAR-100 dataset is collected and proposed by the same researchers, and both dataset have same image size and number of channels. However, the CIFAR-100 dataset has up to 100 image categories, including 20 classifications (superclasses) such as household appliances, furniture, and insects. Each classification can be further divided into five subcategories; for example, the household appliance classification consisting of clocks, keyboards, lamps, phones, and TVs. Although the image shapes are the same as those of the CIFAR-10 dataset, the significantly increased number of image categories makes the training of the neural network model on the CIFAR-100 dataset much more complex than on the CIFAR-10 dataset. Thus, an increase in the number of neural network layers is required to achieve sufficient prediction performance on the CIFAR-100 dataset. In this study, the Dirichlet function is used to partition the dataset to meet various heterogeneous data situations of clients, and the variable α is used to control the non-IID degree in the Dirichlet function. When $\alpha \to 0$, it indicates that the generated dataset has a severe non-IID degree, and when $\alpha \to 1$, it indicates that the generated dataset tends to have an ideal distribution of IID. The federated learning-related parameters are listed in Table 2.

Table 2. The Federated learning parameters.

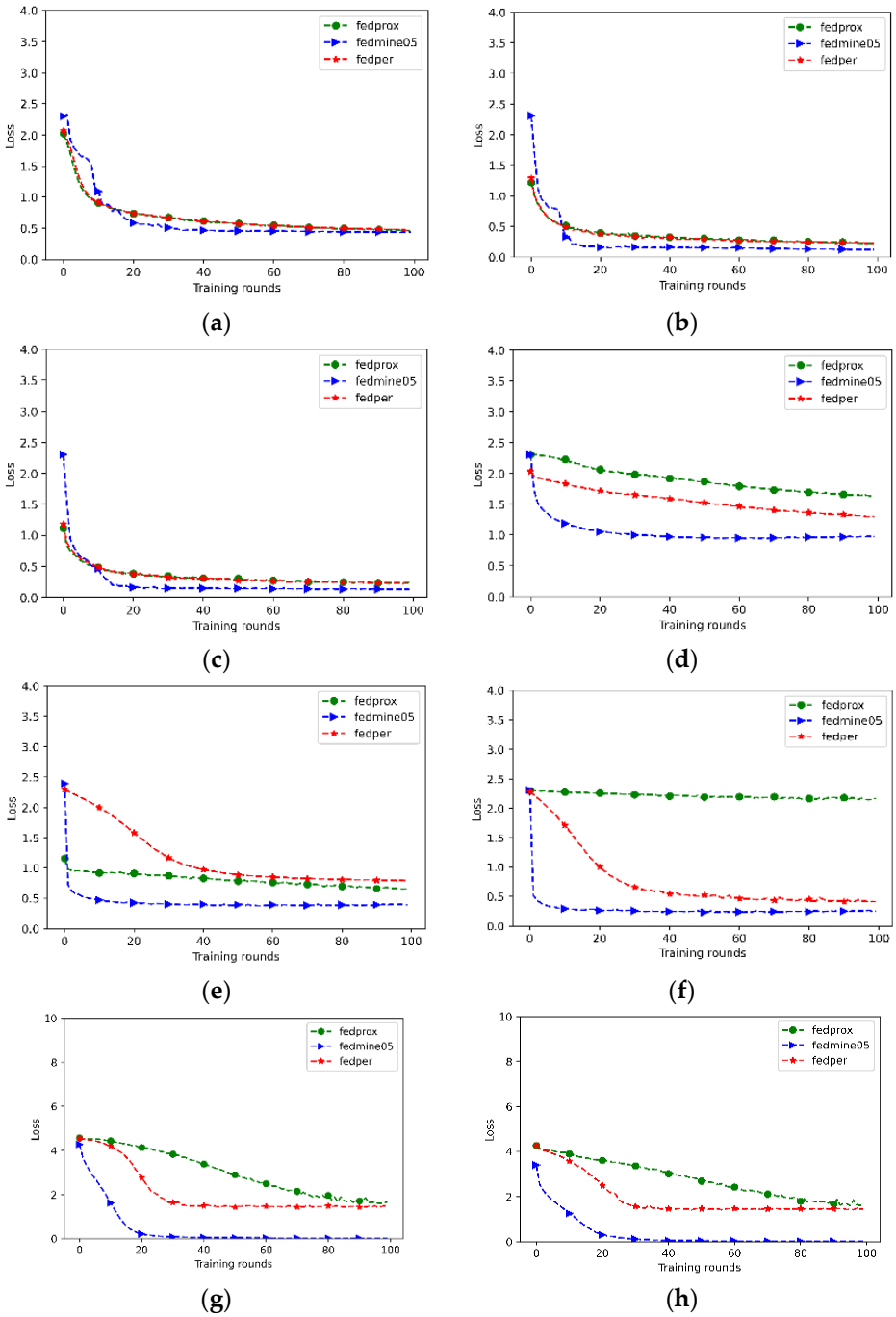
Parameters\Dataset	**Fashion-MNIST**	**CIFAR10**	**CIFAR100**
Input Shape	(28, 28, 1)	(32, 32, 3)	(32, 32, 3)
CNN Layer	2	2	3
FCNN Layer	1	1	2
Federated round	100	100	100
Participating clients	10	10	10
Participating fraction	0.8	0.8	0.8
Client training epoch	1	1	3
Data batch size	32	32	32

4.2. Experimental Results

This study analyzed and compared the proposed federated learning algorithm with the FedProx [18] and FedPer algorithms [21] under different data distributions for three different datasets. The corresponding federated learning loss values are shown in Figure 4. Based on the experimental results, the federated learning algorithm proposed in this study has a lower training loss when facing

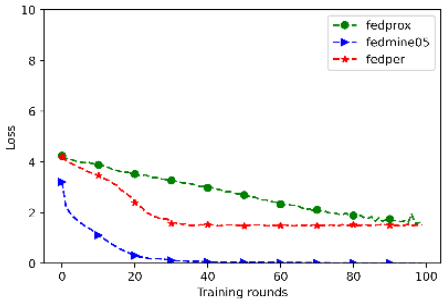
 doi:10.20944/preprints202401.1769.v1

a non-IID data distribution among the client datasets. In the Fashion-MNIST dataset, although the initial β value had a higher loss, it was able to converge within 20 rounds and achieve less data loss. In the CIFAR10 and CIFAR100 datasets, the FedPer federated learning algorithm did not limit the local neural network model training based on the heterogeneity of the client data distribution during the model aggregation stage. This resulted in poor model performance, which became more apparent as the non-IID degree of the dataset increased. When the non-IID degree of the dataset increased, the differences in the neural network model trends among the client datasets widened. Even if a constraint term was added during the federated learning training process to restrict the client model trends, it could not solve the impact of client drift caused by this data distribution. In contrast, the algorithm proposed in this study reduced the impact of client drift through personalized federated learning, and aggregated the models with limitations based on the premise that clients have their own good models after federated learning ends. This resulted in a better model convergence speed and data loss rate. Therefore, the federated learning algorithm proposed in this study can better reduce the impact of client drift caused by a non-IID data distribution.



(a) (b) (c) (d) (e) (f) (g) (h)

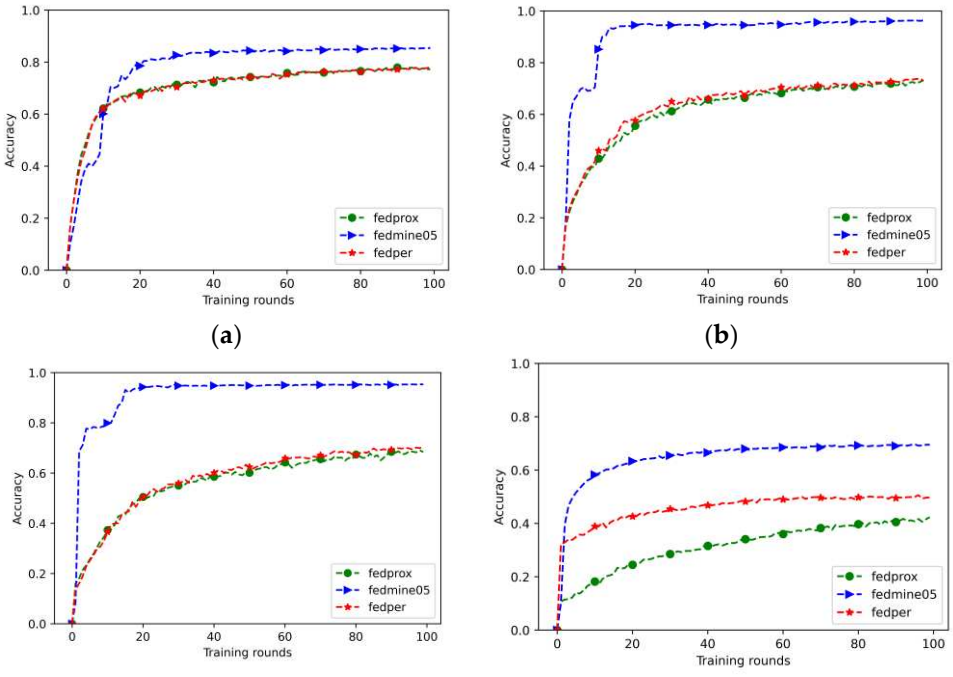
 doi:10.20944/preprints202401.1769.v1



(i)

Figure 4. Loss value in various datasets (a) Fashion-MNIST with α = 1; (b) Fashion-MNIST with α = 0.1; (c) Fashion-MNIST with α = 0.05; (d) CIFAR-10 with α = 1; (e) CIFAR-10 with α = 0.1; (f) CIFAR-10 with α = 0.05; (g) CIFAR-100 with α = 1; (h) CIFAR-100 with α = 0.1; (i) CIFAR-100 with α = 0.05.

The prediction accuracy results for the three datasets are shown in Figure 5. For the Fashion-MNIST dataset, the accuracy of the proposed method was higher than those of the other two federated learning algorithms. It can be found that as the distribution of each dataset becomes more decentralized, the accuracy of both the FedProx and FedPer algorithms decreases from 74% to about 62%. However, the accuracy rate of the proposed algorithm converges to 94% over 20 rounds. The same 10 image categories as Fashion-MNIST are considered from CIFAR-10 dataset. However, the CIFAR-10 dataset contains colored images. The accuracy of this study was higher than that of the other two algorithms, and as the characteristics of the respective datasets tended to be more personalized, the accuracy gradually increased to 92%. The FedProx algorithm mainly emphasizes improving the local loss function and adjusting the epochs of each client without extracting the feature layer of each client. Consequently, the accuracy rate was less than 10% for the multicategory non-IID dataset. In the CIFAR-100 dataset, the accuracy of this study surpassed that of the FedProx and FedPer algorithms. This is because of the simulation environment of 10 clients, although the maximum convergence accuracy is limited to approximately 57%. Based on accuracy results, it was determined that this algorithm could prevent the decline in the overall model accuracy caused by client drift in a non-IID dataset by truncating the client characteristics and category classification.



(a) (b)

(c) (d)

 doi:10.20944/preprints202401.1769.v1

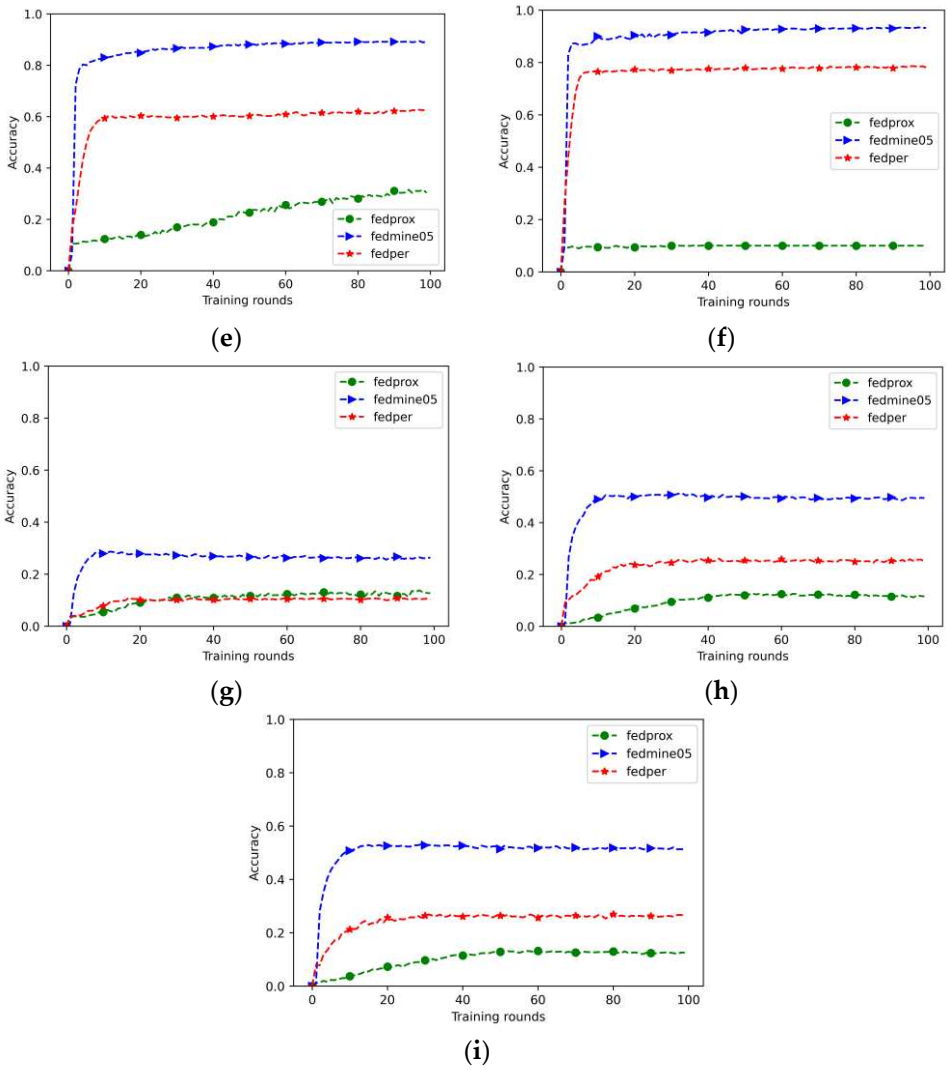


Figure 5. The accuracy rates for various datasets (a) **Fashion-MNIST with** $\alpha = 1$; (b) **Fashion-MNIST with** $\alpha = 0.1$; (c) **Fashion-MNIST with** $\alpha = 0.05$; (d) **CIFAR10 with** $\alpha = 1$; (e) **CIFAR10 with** $\alpha = 0.1$; (f) **CIFAR10 with** $\alpha = 0.05$; (g) **CIFAR100 with** $\alpha = 1$; (h) **CIFAR100 with** $\alpha = 0.1$; (i) **CIFAR100 with** $\alpha = 0.05$.

5. Conclusions

This study proposed a personalized federated learning algorithm based on the concept of multi-task learning, which divides client models into feature extraction layers and category prediction layers. The model category prediction layer is trained using only the client data to maintain the characteristics of each client data, whereas the feature extraction layer is trained using all clients. In addition, model aggregation is limited based on the difference between the previous round client model and the global model. In this study, three image datasets, Fashion-MNIST, CIFAR-10, and CIFAR-100, were used, and the data were divided into non-IID datasets of different degrees. The experimental results showed that under a non-IID data distribution, the proposed federated learning algorithm converges within five rounds and achieves a 0.5 data loss on the CIFAR-10 dataset. It converges within 20 rounds and achieves a 0.1 data loss on the CIFAR-100 datasets. The proposed method has a faster convergence speed and lower data loss than the FedProx and FedPer federated learning algorithms for all three datasets, demonstrating its effectiveness in non-IID data distributions.

Overall, this study provides a new federated learning algorithm that can effectively train deep learning models while protecting personal privacy. However, category prediction is modeled by the distribution of the number of data categories for each client without considering the relevant data

features and number differences. The results of this study are mainly limited to the distribution of classes with similar data features and cannot be inferred when the data features are very different. In the future, the performance evaluations of diverse client data with heterogeneity could be an extension of this study.

References

1. Chung, Y.-L. Application of an Effective Hierarchical Deep-Learning-Based Object Detection Model Integrated with Image-Processing Techniques for Detecting Speed Limit Signs, Rockfalls, Potholes, and Car Crashes. Future Internet 2023, 15, 322. https://doi.org/10.3390/fi15100322.
2. Yen, Chih-Ta, and Guan-Yu Chen. A Deep Learning-Based Person Search System for Real-World Camera Images. Journal of Internet Technology, 2022, 23, 4, pp. 839-851.
3. Ma, Y. W., Chen, J. L., Chen, Y. J., & Lai, Y. H. Explainable deep learning architecture for early diagnosis of Parkinson's disease. Soft Computing, 2023,27, 5, pp. 2729-2738. https://doi.org/10.1007/s00500-021-06170-w.
4. Prasetyo, H., Prayuda, A. W. H., Hsia, C. H., & Wisnu, M. A. Integrating Companding and Deep Learning on Bandwidth-Limited Image Transmission. Journal of Internet Technology, 2022,23, 3, pp. 467-473.
5. Li, Q., Wen, Z., Wu, Z., Hu, S., Wang, N., Li, Y., Liu Xu, & He, B. A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection. IEEE Transactions on Knowledge and Data Engineering, 2023, 35, 4, pp. 3347-3366, https://doi.org/10.1109/TKDE.2021.3124599.
6. Lyu, L., Yu, H., Zhao, J., Yang, Q. Threats to Federated Learning. In: Yang, Q., Fan, L., Yu, H. (eds) Federated Learning. Lecture Notes in Computer Science(), 2020, 12500. Springer, Cham. https://doi.org/10.1007/978-3-030-63076-8_1
7. Tan, Y., Long, G., Liu, L., Zhou, T., Lu, Q., Jiang, J., & Zhang, C. Fedproto: Federated prototype learning across heterogeneous clients. In Proceedings of the AAAI Conference on Artificial Intelligence, 2022, 36, 8, pp. 8432-8440.
8. Xu, J., Glicksberg, B. S., Su, C., Walker, P., Bian, J., & Wang, F. Federated learning for healthcare informatics. Journal of Healthcare Informatics Research, 2021, 5, pp. 1-19.
9. Zheng, Z., Zhou, Y., Sun, Y., Wang, Z., Liu, B., & Li, K. Applications of federated learning in smart cities: recent advances, taxonomy, and open challenges. Connection Science, 2022, 34, 1, pp. 1-28.
10. Nikolaidis, F.; Symeonides, M.; Trihinas, D. Towards Efficient Resource Allocation for Federated Learning in Virtualized Managed Environments. Future Internet 2023, 15, 261. https://doi.org/10.3390/fi15080261
11. Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. Federated learning: Challenges, methods, and future directions. IEEE signal processing magazine, 2020, 37, 3, pp. 50-60.
12. Singh, P., Singh, M. K., Singh, R., & Singh, N. Federated learning: Challenges, methods, and future directions. In Federated Learning for IoT Applications. Cham: Springer International Publishing, 2022, pp. 199-214.
13. McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A.. Communication-efficient learning of deep networks from decentralized data. In Artificial intelligence and statistics, 2017, pp. 1273-1282.
14. Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., & Suresh, A. T. Scaffold: Stochastic controlled averaging for federated learning. In International conference on machine learning, 2020, pp. 5132-5143.
15. Karimireddy, S. P., Jaggi, M., Kale, S., Mohri, M., Reddi, S. J., Stich, S. U., & Suresh, A. T. Mime: Mimicking centralized stochastic algorithms in federated learning. arXiv preprint arXiv:2008.03606. 2020.
16. Wang, H., Kaplan, Z., Niu, D., & Li, B. Optimizing federated learning on non-iid data with reinforcement learning. In IEEE INFOCOM 2020-IEEE Conference on Computer Communications, 2020, pp. 1698-1707.
17. Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., & Chandra, V. Federated learning with non-iid data. arXiv preprint arXiv:1806.00582. 2018.
18. Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., & Smith, V. Federated optimization in heterogeneous networks. Proceedings of Machine learning and systems, 2020, 2, pp. 429-450.
19. Wu, Q., He, K., & Chen, X. Personalized federated learning for intelligent IoT applications: A cloud-edge based framework. IEEE Open Journal of the Computer Society, 2020, 1, pp. 35-44.
20. Li, T., Hu, S., Beirami, A., & Smith, V. Ditto: Fair and robust federated learning through personalization. In International Conference on Machine Learning, 2021, pp. 6357-6368.
21. Arivazhagan, M. G., Aggarwal, V., Singh, A. K., & Choudhary, S. Federated learning with personalization layers. arXiv preprint arXiv:1912.00818. 2019.
22. Zhang, Y., & Yang, Q. An overview of multi-task learning. National Science Review, 2018, 5, 1, pp. 30-43.
23. Duchi, J., Hazan, E., & Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. Journal of machine learning research, 2011, 12, 7
24. Deng, Y., Kamani, M. M., & Mahdavi, M. Adaptive personalized federated learning. arXiv preprint arXiv:2003.13461. 2020.

25. Beutel, D. J., Topal, T., Mathur, A., Qiu, X., Fernandez-Marques, J., Gao, Y., Sani, L., Li, K.H., Parcollet, T., de Gusmão, P.P.B. & Lane, N. D. Flower: A friendly federated learning research framework. arXiv preprint arXiv:2007.14390. 2020.
26. Li, K. H., de Gusmão, P. P. B., Beutel, D. J., & Lane, N. D. Secure aggregation for federated learning in flower. In Proceedings of the 2nd ACM International Workshop on Distributed Machine Learning, 2021, pp. 8-14.
27. Brum, R., Drummond, L., Castro, M. C., & Teodoro, G. Towards optimizing computational costs of federated learning in clouds. In 2021 International Symposium on Computer Architecture and High Performance Computing Workshops (SBAC-PADW), 2021, pp. 35-40.
28. Xiao, H., Rasul, K., & Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747. 2017.
29. Krizhevsky, A., & Hinton, G. Learning multiple layers of features from tiny images. 2009.
30. Li, Hongxia and Cai, Zhongyi and Wang, Jingya and Tang, Jiangnan and Ding, Weiping and Lin, Chin-Teng and Shi, Ye, "FedTP: Federated Learning by Transformer Personalization," in IEEE Transactions on Neural Networks and Learning Systems, doi: 10.1109/TNNLS.2023.3269062.
31. Zhai, R.; Chen, X.; Pei, L.; Ma, Z. A Federated Learning Framework against Data Poisoning Attacks on the Basis of the Genetic Algorithm. Electronics 2023, 12, 560. https://doi.org/10.3390/electronics12030560