Received 1 May 2024; revised 2 July 2024; accepted 16 July 2024. Date of publication 29 July 2024; date of current version 10 September 2024.

Digital Object Identifier 10.1109/OJCOMS.2024.3435389

FedCPD: A Federated Learning Algorithm for Processing and Securing Distributed Heterogeneous Data in the Metaverse

LE SUN [1], ZHIMENG ZHANG [1], AND GHULAM MUHAMMAD [2] (Senior Member, IEEE)

[1] Department of Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology, Nanjing University of Information Science and Technology, Nanjing 210044, China

[2] Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11451, Saudi Arabia

CORRESPONDING AUTHOR: G. MUHAMMAD (e-mail: ghulam@ksu.edu.sa)

This work was supported by the Researchers Supporting Project King Saud University, Riyadh, Saudi Arabia, under Grant RSP2024R34.

(Special Issue on Challenges and Opportunities in Metaverse-Based Communication and Networking)

ABSTRACT The continuous development of virtual reality technology allows the metaverse to create more immersive and highly interactive experiences for users. Metaverse users upload personal information through virtual reality devices, causing data security and communication security issues. Moreover, the diversity of data sources within the metaverse exacerbates issues of data heterogeneity. To address these issues, we propose a generative learning-based federated learning algorithm to secure and process heterogeneous data from users in the metaverse, called FedCPD. It consists of three main modules: a privacy protection module for data security, a correction module to correct the bias of the classifier, and an aggregation module to improve model performance. To protect the data security of metaverse users, we design a privacy-preserving method based on conditional Generative Adversarial Networks (cGAN) in the privacy protection module. The method replaces the feature extractor with a generator in cGAN to engage in server-side aggregation to avoid data exposure. The correction module is proposed to enhance the classifier's ability to classify unknown data by using the constructed pseudo dataset for classification model training. To alleviate the negative impact of data heterogeneity on the global model, the aggregation module utilizes local discrepancy-based aggregation weights for server-side aggregation. It assigns higher aggregation weights to the client models that perform better than other models. Extensive experiments on multiple datasets show that FedCPD exhibits the highest classification accuracy compared to existing algorithms, demonstrating its effectiveness in processing heterogeneous data.

INDEX TERMS Metaverse, federated learning, generative learning, data security, data heterogeneity.

I. INTRODUCTION

METAVERSE signifies a digitized world beyond reality, encompassing various technologies and applications [1]. It can be regarded as a carrier of big data and information technology. In the metaverse, users interact through local extended reality devices, leading to inevitable communication security concerns [2]. Additionally, if user data is uploaded to the server for processing, it will result in significant communication overhead and server computational pressure [3]. Edge Intelligence is a new computing paradigm that pushes intelligent computing capabilities toward the network edge [13]. In the context of computing edge-cloud continuum, this method enables local processing and analysis of data, significantly reducing the time taken to transmit data from edge devices to the cloud, thereby reducing latency and enhancing real-time responsiveness. It also effectively conserves bandwidth resources and strengthens data security and privacy protection. Furthermore, recent research has focused on utilizing Edge Artificial Intelligence (AI) techniques to handle and analyze data on edge devices [40]. This paradigm involves not only running artificial intelligence models on edge devices but also encompasses collaborative computing and data processing between edge devices and the cloud. Federated Learning

(FL) is a distributed machine learning paradigm that can provide strong support for Edge AI, effectively addressing these two issues [4].

Within the framework of FL, users in the metaverse engage in collaborative training of a global model without necessitating the direct upload of their individual datasets [5]. They can leverage the private data residing within their virtual reality devices to conduct local model updates. These updated model parameters are then transmitted to a cloud server for aggregation, thereby facilitating the enhancement of the global model's performance. Although this decentralized training method prevents servers from directly accessing private data, the extractor based on private data still poses a risk of exposure to the server. Recent research has shown that such a method is vulnerable to privacy attacks, such as Property Inference Attacks and Reconstruction Attacks [6], [7]. These attacks engender the compromise of user privacy, thereby posing a substantive peril to the data security of metaverse participants [38]. To address this issue, we utilize generators from conditional generative adversarial networks (cGAN) instead of feature extractors for aggregation. We employ a local optimization strategy to minimize the discrepancy between the output distributions of the feature extractor and the generator. It serves to safeguard the privacy of local data while facilitating the acquisition of shared knowledge by the feature extractor.

In the metaverse, data from different clients often exhibit non-independent and identically distributed (non-IID) characteristics [8]. This phenomenon arises from the heterogeneity in feature and category distributions of virtual data and user information across distinct virtual reality devices utilized by various users [17]. The imbalance in datasets across clients leads to imbalanced model training and poorer performance of the aggregated model, thereby reducing the effectiveness of processing non-IID data in the metaverse through FL [9]. The research suggests that in FL, clients whose local data distribution closely matches the global data distribution tend to generate better-performing local models [10]. Therefore, we employ discrepancy-based aggregation weights (DBAW) to aggregate local models to enhance users' access to more accurate classification results. It allocates larger aggregation weights to local models with better performance during aggregation, thereby endowing the aggregated model with superior performance.

Furthermore, the future data of users in the metaverse is unknown. Pre-trained local models may exhibit lower classification accuracy when confronted with new, unseen data [11]. This phenomenon emerges due to the inherent bias introduced within the classifiers of local classification models, which is a consequence of their adaptation to local data sets [12]. Therefore, we generate a globally shared, label-independent pseudo dataset from a subset of data within local devices in the metaverse. We utilize the generated pseudo dataset for training local classification models, thereby enhancing the classifiers' ability to classify unknown data. This enables users' local models in the metaverse environment to better handle unseen data and be more robust.

This paper proposes FedCPD, a generative learning-based FL algorithm to secure and process heterogeneous data from users in the metaverse. In contrast to alternative FL frameworks, FedCPD utilizes generators in cGAN instead of extractors to participate in server-side aggregation to secure metaverse user data. Moreover, our algorithm calculates the discrepancy between the distributions of client data and the global distribution. Leveraging this discrepancy, we introduce DBAW, wherein better-performing local classification models are accorded greater aggregation weights. Additionally, we design a correction module. Specifically, it utilizes a globally shared, label-independent pseudo dataset to mitigate biases within local classifiers, thereby enhancing their classification efficacy on previously unseen data. Our contributions can be summarized as follows:

- We propose FedCPD, a secure and high-performance generative learning-based FL algorithm. It secures metaverse user data through a cGAN-based privacy-preserving module.
- We design a correction module to correct the classifier with a globally shared, label-independent pseudo dataset. It enhances the ability of the local classifier to classify unknown data in the metaverse.
- To mitigate the impact of data heterogeneity on model performance, we design an aggregation module to assign higher aggregation weights to the client models that perform better than other models.
- Extensive experiments on six datasets validate the effectiveness of FedCPD. Our proposed algorithm achieves the highest classification accuracy across all datasets.

The rest of the paper is structured as follows: Section II introduces related work. Section III provides a detailed description of the FedCPD. Section IV presents comprehensive experiments and results. Section V summarizes all the work.

II. RELATED WORK

A. DATA SECURITY IN THE METAVERSE

Metaverse is a parallel digital space that coexists with the real world, integrating social, immersive interaction, and scalability. However, alongside its advantages, the metaverse also poses risks to security and privacy, such as personal information leakage, eavesdropping, and data theft [2]. In recent years, ensuring data security in the metaverse has attracted widespread attention from researchers.

Thakur et al. [14] proposed a secure mutual authentication scheme utilizing elliptic curve cryptography and fuzzy extractors to address security attacks like replay and impersonation in the metaverse. Yang et al. [15] utilized a decentralized authentication protocol based on avatar identity models and chameleon collision signatures to achieve real-time authentication of avatar identities. It ensures virtual-physical traceability within the metaverse, enabling the tracking of malicious actors in the physical

TABLE 1. Comparison with the related works.

Paper	Calculation of aggregation weights	Correction of local classifiers	Generative learning-based	Data imbalance
[14]	-	No	No	No
[15]	-	No	No	No
[16]	-	No	No	No
[18]	Datasize	No	No	Yes
[19]	Datasize	No	Yes	Yes
[8]	Datasize	No	No	Yes
[11]	Datasize	Yes	No	Yes
FedCPD	LCD + Datasize	Yes	Yes	Yes

world through avatars in virtual space. Therefore, this approach somewhat reduces the number of attackers who jeopardize the data privacy of metaverse users. Li et al. [16] introduced a secure communication model based on semantic blocks, semantic variable encoding, and hybrid channels with concealed tasks to enhance the security of semantic communication in the metaverse. This method not only increases the communication density in the metaverse but also mitigates the challenge of black-box attacks on metaverse users during communication.

B. FEDERATED LEARNING FOR NON-IID DATA IN THE METAVERSE

Data imbalance and heterogeneity are widely present in the metaverse, encompassing differences in features and distribution of categories across different virtual reality devices [8]. In recent years, FL has shown tremendous potential in handling non-IID data [39]. It can facilitate collaboration among different clients to train unique local models while preserving the privacy of metaverse user data.

Zhou et al. [18] proposed a personalized FL framework incorporating model contrastive learning. This framework achieves effective fusion of non-IID data in the metaverse by constructing a personalized multimodal fusion network. Chen et al. [19] designed a trustworthy semantic communication system for the metaverse based on the FL architecture, which enables effective distributed decision-making and privacy protection. Additionally, the system utilizes low-speed semantic communication to support the metaverse, thus circumventing the bottleneck of limited communication and computational resources in the metaverse. Zeng et al. [8] leveraged FL within the industrial metaverse, mitigating data heterogeneity through dynamic grouping and training mode transformation. This methodology tackles challenges including learning degradation due to non-IID data and constraints posed by limited communication bandwidth. Guo et al. [11] addressed the challenges arising from local updates in supervised FL by reducing local learning biases in features and classifiers. However, these methods overlook the impact of local category discrepancy (LCD) on aggregation weights during server-side aggregation, resulting in suboptimal performance of the aggregated global model. In addition, since these methods involve uploading the entire model parameters, especially the feature extractors that are in direct contact with the data, to the server during communication, local user data becomes highly susceptible to exposure.

The comparison between FedCPD and related works is summarized in Table 1. Our proposed framework employs DBAW and a method of correcting local classifiers using a globally shared, label-independent pseudo dataset. Compared to the aforementioned methods, FedCPD takes into account the impact of local class distribution on aggregation weights and the bias introduced by local classifiers adapting to local datasets. Therefore, FedCPD can more effectively address the impact of data heterogeneity on model accuracy in the metaverse. Additionally, FedCPD utilizes cGAN to retain the feature extractor that directly interacts with user data locally, which can prevent user data from being exposed to the server and leaks during communication

III. METHODOLOGY

A. FRAMEWORK OVERVIEW

We propose a generative-learning based FL algorithm called FedCPD to secure and process heterogeneous data in the metaverse. Each virtual reality apparatus within the metaverse is regarded as a client. To simulate data heterogeneity in the metaverse scenario, we use both naturally imbalanced datasets and datasets partitioned by Dirichlet distribution. Detailed implementation and analysis of these methods will be elucidated in Section IV.

As shown in Fig. 1, FedCPD consists of five key steps. 1) We construct a globally shared, label-independent pseudo dataset using a subset of the local data. Each client receives global model parameters from the server and initializes its local model. 2) The local model is trained using local data and our correction module uses the constructed pseudo dataset to correct the classifier. 3) Each client computes the local category discrepancy by comparing the local data distribution with the global distribution. 4) The server collects the locally computed LCD and dataset sizes from the clients to calculate aggregation weights. 5) Utilizing DBAW, the server aggregates the models uploaded by each client and updates the global model through knowledge distillation. The specific implementation process is given in the following five subsections of the section.

Assuming there are n clients in the metaverse scenario. Each client has its own local dataset $\{X_1, X_2, \ldots, X_n\}$, as

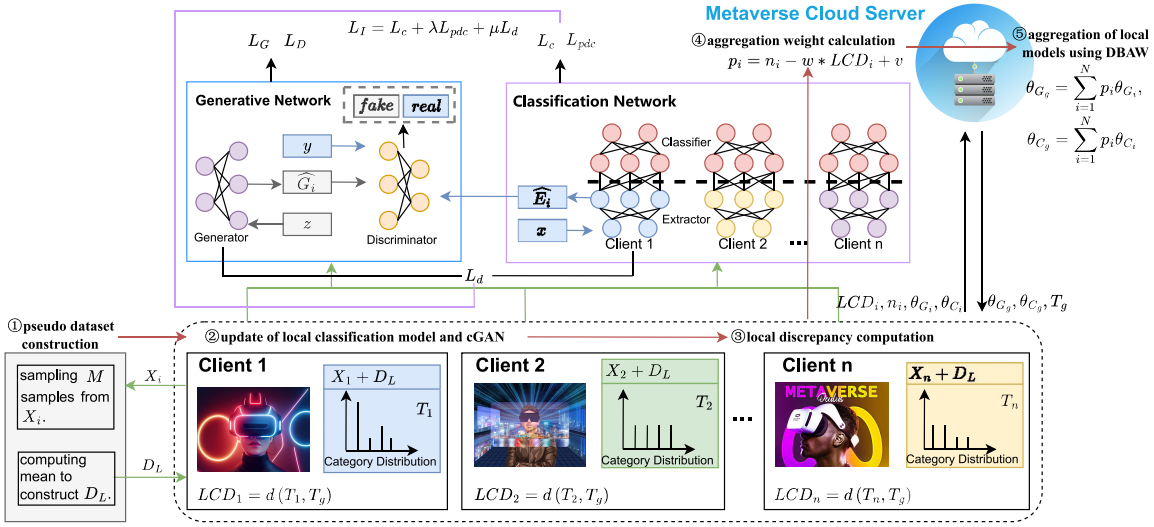


FIGURE 1. The overview of FedCPD.

well as a local classification model and a cGAN. On client i, the classification model consists of a feature extractor E_i and a classifier C_i, while the cGAN consists of a generator G_i and a discriminator D_i. During local training, we use noise z and labels y to train the cGAN, so that the output distribution of the cGAN's generator is similar to the output distribution of the feature extractor when the input is real data x.

Throughout the communication rounds of FL, client i utilizes its local data and pseudo dataset to locally train both the classification model and the cGAN. It also computes the discrepancy between its local class distribution and the global distribution. Subsequently, client i uploads C_i from the classification model and G_i from the cGAN, along with LCD_i, to the server. In contrast, E_i from the classification model and D_i from the cGAN are retained locally to preserve privacy. Upon receiving the locally uploaded information, the server computes an aggregation weight p_i for each client. Utilizing these aggregation weights and knowledge distillation, the server proceeds to perform aggregation, thereby constructing the global classifier C_g and the global generator G_g.

B. PSEUDO DATASET CONSTRUCTION

In this paper, we use two methods to construct pseudo dataset. Firstly, we utilize a subset of the local dataset to construct the pseudo dataset, Algorithm 1 presents the method for generating pseudo dataset. Specifically, we randomly draw M samples on client i and locally compute its mean as a pseudo dataset sample, denoted as $\widetilde{x_L}$, and set its label to $\widetilde{y_L}$ (Alg. 1, lines 4-6). We call this method Local Data Mean (LDM). Equation (1) demonstrates the samples and labels constructed through LDM.

$$\widetilde{x_L} = \frac{1}{M}\sum_{m=1}^{M} x_m, \quad \widetilde{y_L} = \frac{1}{C} * 1 \tag{1}$$

Algorithm 1 Pseudo Data Construction by LDM

Input: datasets $\{X_1, X_2, \ldots, X_N\}$; sample size M; number of pseudo data for each client P;

1: Initialize $D_L = \emptyset$.
2: **for** $i = 1, \ldots, N$ **do**
3: **for** $p = 1, \ldots, P$ **do**
4: Randomly sample $x_1, \ldots, x_M$ from X_i.
5: $\widetilde{x_L} = \frac{1}{M}\sum_{m=1}^{M} x_m$, $\widetilde{y_L} = \frac{1}{C} * 1$
6: $D_L = D_L \cup \widetilde{x_L}$
7: **end for**
8: **end for**

where C represents the number of classes in the local dataset of each client. The constructed pseudo dataset we denote as D_L.

Secondly, to protect user data privacy, we utilize a globally dataset D_g to construct pseudo dataset locally on the client-side. We call this method Global Data Mean (GDM). This method prevents the exposure of clients' private local data while minimizing the divergence between the distribution of the constructed pseudo dataset and that of the local dataset. The samples $\widetilde{x_{G_i}}$ and corresponding labels $\widetilde{y_{G_i}}$ in the pseudo dataset D_{G_i} at client i are given by Equation (2).

$$\widetilde{x_{G_i}} = \frac{1}{T+1}\left(x_L + \sum_{t=1}^{T} x_t\right),$$
$$\widetilde{y_{G_i}} = \frac{1}{T+1}\left(\frac{1}{C} * 1 + \sum_{t=1}^{T} y_t\right) \tag{2}$$

where T is a constant used to control the similarity between the pseudo dataset and the local data. x_t and y_t are the sample and label of the client's local data, respectively. x_L denotes an LDM sample from the global dataset D_g.

Algorithm 2 Local Update of FedCPD

Input: datasets $\{X_1, X_2, \ldots, X_N\}$; Pseudo dataset D_L; learning rate η_c, η_g; local training epoch E_1.

1: Initialize θ_{E_i}, θ_{D_i}, θ_{G_g}, and θ_{C_g} at random.
2: **for** $i = 1, \ldots, N$ **do**
3: Client i receives θ_{G_g} and θ_{C_g} from the server.
4: $\theta_{G_i} \leftarrow \theta_{G_g}$, $\theta_{C_i} \leftarrow \theta_{C_g}$
5: **Classification Network Update:**
6: **for** $t \in \{1, \ldots, E_1\}$ **do**
7: **for all** $x, y \in X_i \cup D_L$ **do**
8: sample z from $N(0, 1)$
9: $\theta_{E_i} \leftarrow \theta_{E_i} - \eta_c \nabla_{\theta_{E_i}} L_I(x, y, z)$
10: $\theta_{C_i} \leftarrow \theta_{C_i} - \eta_c \nabla_{\theta_{C_i}} L_I(x, y, z)$
11: **end for**
12: **end for**
13: **Generative Network Update:**
14: **for** $t \in \{1, \ldots, E_1\}$ **do**
15: **for** $x, y \in X_i$ **do**
16: sample z from $N(0, 1)$
17: $\theta_{G_i} \leftarrow \theta_{G_i} - \eta_g \nabla_{\theta_{G_i}} L_G(z, y)$
18: $\theta_{D_i} \leftarrow \theta_{D_i} - \eta_g \nabla_{\theta_{D_i}} L_D(x, z, y)$
19: **end for**
20: **end for**
21: sends θ_{G_i} and θ_{C_i} to the server.
22: **end for**

C. UPDATE OF LOCAL CLASSIFICATION MODEL AND CGAN

Alg. 2 outlines the entire process of the local update. In the phase of updating the local classification model, client i optimizes the classification model by minimizing the classification loss over both its local data and the pseudo dataset (Alg. 2, lines 6-12). Equation (3) presents the classification loss function for the local data at each client i.

$$L_c = E_{x,y\sim X_i} CEL\big(\hat{y}, y\big) \tag{3}$$

where CEL represents the cross-entropy loss function, and $\hat{y}$ is a probability vector obtained by inputting data x into the local classification model. Due to label distribution skew or the absence of certain samples from majority/minority classes, the classifier of the locally trained model often overfits the categories present locally. To address this, we use the pseudo dataset constructed in Section III-B to emulate the global data distribution, eliminating classifier bias by enforcing an even output distribution for the pseudo dataset. The loss function for the pseudo dataset on the classification model is given by Equation (4).

$$L_{pdc} = E_{x,y\sim D_L} CEL\big(\hat{y}, y\big) \tag{4}$$

Furthermore, during local training, client i incorporates the shared knowledge from the previous round's global generator G_g into its local feature extractor E_i. To achieve this, we keep the parameters of the global generator fixed and optimize its feature extractor by minimizing the mean squared error loss. We denote the output of the feature extractor E_i and the global generator G_g as $F_i = E_i(x|y)$ and $F_g = G_g(z, y)$, respectively, with the loss function for the feature extractor being presented in Equation (5).

$$L_d = E_{x,y\sim X_i} E_{z\sim N} \big|F_i - F_g\big|^2 \tag{5}$$

where N denotes the distribution of random noise. Having established the loss functions for the local data classification, pseudo dataset classification, and feature extractor, we employ Equation (6) as the overall loss function for the local classification model update phase.

$$L_I = L_c + \lambda L_{pdc} + \mu L_d \tag{6}$$

where λ and μ are non-negative hyperparameters used to balance the three loss functions. Since in the early stages of training, the global generator G_g is unable to accurately generate features, we set μ to 0. As the training iterations progress and the generator becomes more adept at fitting the feature extractor, μ is increased from 0 to 1.

During the local cGAN update phase (Alg. 2, lines 14-20), the objective for each client i is to make the output of its local generator G_i as close as possible to the output of its local feature extractor E_i. To achieve this goal, E_i's parameters are first kept fixed, and then the cGAN's generator G_i is trained. Specifically, we randomly select a mini-batch of data containing samples x and corresponding labels y. The samples are input into the feature extractor to obtain the actual output e. Next, we generate noise z of the same batch size and feed it, along with the labels y, into the generator G_i to produce the approximate output $\hat{e}$. Using both the actual feature output e and the approximate feature output $\hat{e}$, we calculate the discriminator loss L_d and generator loss L_g by feeding them into the discriminator D_i. We update G_i and D_i by minimizing these two losses.

We denote the discriminator's prediction for the generator's output as $D_{gen} = D_i(G_i(z, y; \theta_{G_i}); \theta_{D_i})$, and its prediction for the feature extractor's output as $D_{ext} = D_i(E_i(x|y; \theta_{E_i}); \theta_{D_i})$. Equations (7) and (8) present the discriminator loss L_D and generator loss L_G, respectively.

$$L_D = E_{x,y\sim X_i} E_{z\sim N} \big(\log(1 - D_{ext}) + \log D_{gen}\big) \tag{7}$$

$$L_G = E_{x,y\sim X_i} E_{z\sim N} \log\big(1 - D_{gen}\big) \tag{8}$$

Through these two loss functions, we can effectuate the updates of the generator G_i and discriminator D_i at each client i. Upon completion of local training, each client i uploads their respective local generator G_i and classifier C_i to the server for aggregation.

D. LOCAL DISCREPANCY COMPUTATION

Each client is required to locally compute the discrepancy between their local class distribution and the global class distribution. However, directly transmitting the local class distribution to the server raises privacy concerns due to potential exposure. To address this issue, we employ secure aggregation [20], enabling clients to send their respective

class distributions T_i to the server without revealing individual distributions. The server can then compute the actual global class distribution T_g without knowledge of each client's specific distribution.

In more detail, for any two clients C_i and C_j, a mutually determined random vector $V_{i,j}$ is generated between them. The relative size of the dataset, denoted as n_i is given by Equation (9).

$$n_i = \frac{|X_i|}{\sum_{j=1}^{N} |X_j|} \tag{9}$$

where $|X_i|$ represents the total number of samples contained in client i. For client i, its local class distribution can be transformed into the following form as shown in Equation (10).

$$\widetilde{T}_i = n_i T_i + \sum_{j=1}^{i-1} V_{i,j} - \sum_{j=i+1}^{n} V_{i,j} \tag{10}$$

After each client i uploads their local $\widetilde{T}_i$ to the server as per Equation (11), the server can perform aggregation to obtain the global class distribution $\widetilde{T}_g$.

$$\begin{aligned} T_g &= \sum_{i=1}^{n} \widetilde{T}_i \\ &= \sum_{i=1}^{n} \left(n_i T_i + \sum_{j=1}^{i-1} V_{i,j} - \sum_{j=i+1}^{n} V_{i,j} \right) \\ &= \sum_{i=1}^{n} n_i T_i \end{aligned} \tag{11}$$

Moreover, since the distribution discrepancies are computed solely on the client-side, this significantly reduces the risk of local class distribution leakage. We denote the local class distribution discrepancy at client i as LCD_i. Given the global distribution T_g and the local distribution T_i, each client i can compute its local difference level by assessing the discrepancy between these two distributions, as expressed in Equation (12).

$$LCD_i = d\big(T_i, T_g\big) \tag{12}$$

where $d(,)$ represents a predefined distance metric (such as the Kullback-Leibler (KL) divergence or the L2 norm). Upon calculating the local difference level, client i uploads LCD_i to the server.

E. GLOBAL MODEL AGGREGATION WEIGHT CALCULATION

In traditional FL, when aggregating local models, weights are often assigned based on the size of each client's dataset. Inspired by FedDisco [10], however, we argue that such weight calculation methods overlook the impact of local class distributions on model classification accuracy. By leveraging the relative dataset size n_i and the local difference level LCD_i, we can assign more discriminative aggregation weights to each client i. The aggregation weight p_i thus determined is given by Equation (13).

$$p_i = \frac{\phi(n_i - w * LCD_i + v)}{\sum_{j=1}^{N} \phi\big(n_j - w * LCD_j + v\big)} \tag{13}$$

where ϕ denotes the ReLU activation function, which handles negative values. Parameters w and v are hyperparameters; the former balances the contributions of n_i and LCD_i, while the latter adjusts the overall magnitude of the weights. This approach assigns larger aggregation weights to clients with larger dataset sizes and smaller local difference levels.

Algorithm 3 Server Aggregation of FedCPD

Input: size of dataset $\{|X_1|, |X_2|, \ldots, |X_n|\}$; learning rate η_s; server training epoch E_2 for knowledge distillation; local category discrepancy: $\{|LCD_1|, |LCD_2|, \ldots, |LCD_N|\}$; sample batch size B.

1: Receives $\{|LCD_i|\}_{i=1}^{N}$, $\{\theta_{G_i}\}_{i=1}^{N}$, and $\{\theta_{C_i}\}_{i=1}^{N}$ from each client.
2: **for** $i = 1, \ldots, N$ **do**
3: computes p_i by Equation (11)
4: **end for**
5: $\theta_{G_g} = \sum_{i=1}^{N} p_i \theta_{G_i}$, $\theta_{C_g} = \sum_{i=1}^{N} p_i \theta_{C_i}$
6: **for** $t \in \{1, \ldots, E_2\}$ **do**
7: Sample (z, y)
8: $\theta_{G_g,\ C_g} \leftarrow \theta_{G_g,\ C_g} - \eta_s \nabla_{\theta_{G_g,\ C_g}} L_{KL}(z, y)$
9: **end for**
10: Sends $\theta_{G_g,\ C_g}$ to each client.

F. SERVER-SIDE AGGREGATION OF LOCAL MODELS

Alg. 3 outlines the entire process on the server side for computing global model aggregation weights and constructing the global generator and classifier through knowledge distillation. The server computes the global model aggregation weights based on the local discrepancies uploaded by users (Alg. 3, lines 2-4). Subsequently, the server employs these weights to perform a weighted aggregation of the G_i and C_i models uploaded by clients, thereby initializing the global generator G_g and classifier C_g.

Regarding the distillation process (Alg. 3, lines 6-9), the server first generates a small batch of training data (z, y), where the labels y are sampled from a uniform distribution $U(0, c)$, and the noise z is sampled from a Gaussian distribution $N(0, 1)$, with c denoting the number of sample categories. The server then feeds (z, y) into all generators G_i and computes the class probability distributions $P_c(y, z)$ and $P_s(y, z)$. The former is derived by passing the data through the local client-side classifiers C_i, while the latter is obtained via the global classifier C_g.

Having obtained both probability distributions, the server updates G_g and C_g by minimizing the KL divergence

between them. The probability distributions are given by equations (14) and (15), respectively.

$$P_c(y,z) = softmax\left(\sum_{i=1}^{n} p_i C_i\left(G_g(y,z)\right)\right) \tag{14}$$

$$P_s(y,z) = softmax\left(C_g\left(G_g(y,z)\right)\right) \tag{15}$$

Equation (16) presents the KL divergence between two distributions.

$$L_{KL} = E_{y\sim U} E_{z\sim N} KL(P_c(y,z), P_s(y,z)) \tag{16}$$

In KL, the server does not require access to any client data, which to some extent ensures data security. Upon completion of aggregation, the server dispatches both the global generator and the global classifier to all clients.

IV. EXPERIMENTS

Our experiments are conducted on a computer with an Intel Core i9-11900K CPU, 32.00 GB of memory, and an NVIDIA GeForce RTX 3090 GPU. The experimental environment includes Python 3.9, PyTorch 1.13.1, and Windows 10.

A. EXPERIMENT SETTING

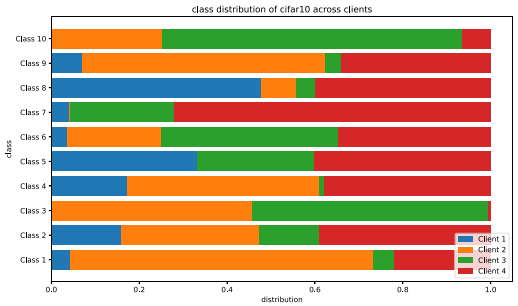
1) DATASETS

We conduct experiments with all FL algorithms across six datasets: Real-world dermoscopic FL dataset (Dermoscopic) [21], [22], Office-Caltech10 (Office) [23], Digit5 [24], DomainNet [24], Fmnist, and Cifar10. Notably, these datasets are of significant value as they can be treated as medical images, office supplies, handwritten digits, etc. from different domains in the metaverse. They are important for evaluating the applicability of FedCPD in the metaverse environment.

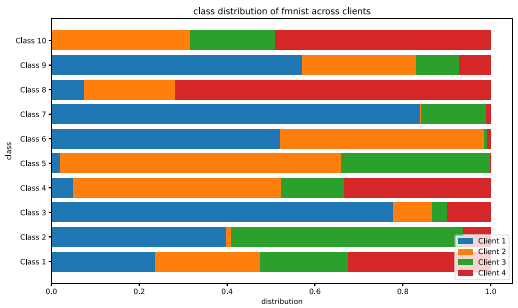
Specifically, a total of 8,940 samples are collected from the HAM10K [22] dataset, along with 2,000 samples from the Memorial Sloan-Kettering Cancer Center (MSK) [21] dataset, to form the skin lesion dataset. These samples can be categorized into three classes: nevi (nv), benign keratosis (bkl), and melanoma (mel). The Dermoscopic encompasses four domains: ViDIR Current, ViDIR MoleMax, Rosendahl, and MSK. Office consists of ten office object classes drawn from four distinct domains: Amazon,[1] DSLR [25], Webcam [25], and Caltech [26]. Digit5 comprises five digit image datasets: MNIST [27], SynthDigits (Syn) [28], MNIST-M [28], Street-View House Number (SVHN) [29], and USPS [30]. DomainNet includes data from six diverse domains: Clipart, Infograph, Painting, Quickdraw, Real, and Sketch. Data in these four datasets originates from different sources, collected via varying methods, inherently conforming to a non-IID setting.

For the IID datasets Fmnist and Cifar10, we partition the datasets according to a dirichlet distribution Dir(α), where α is a parameter controlling the level of heterogeneity in the data division. Under this setting, we allocate 8,000 images among four clients for the training set, with each client's data distribution depicted in Fig. 2.

[1] www.amazon.com



(a)



(b)

FIGURE 2. Cifar10 and Fmnist's data distribution after partitioning based on Dir(α). (a) Cifar10. (b) Fmnist.

TABLE 2. Time and Space complexities.

Model	LeNet-5	Generator	Discriminator
Number of Parameters (K)	62.01	890.62	672.51
Floating Point Operations (M)	0.65	7.95	3.52

2) IMPLEMENTATION DETAILS AND HYPERPARAMETERS

In our experiments, we employ LeNet-5 [27] as the classification model, with the cGAN architecture based on DCGAN [31]. Table 2 shows the time and space complexities of FedCPD.

All experiments utilize the Adam optimizer, with a learning rate of 0.0003 and weight decay set to 0.0001. For each experiment, the global communication rounds are fixed at 150, while local training epoch and server training epoch are set to 20. The batch size is configured to be 16. Training is terminated early if the validation accuracy on the model does not increase after 20 global communication rounds. In Dermoscopic, Office, DomainNet, and Digit5, each domain is treated as a client's dataset. For Dermoscopic, Office, and DomainNet, half of the original data from each domain is designated as the local training set. For Digit5, we randomly select 2,000 images per client for the local training set. For Cifar10 and Fmnist, a total of 8,000 images are allocated to 4 clients for training. Each dataset is experimented upon using five different random seeds, and accuracy is calculated using both the client's validation and test sets.

TABLE 3. Comparison of classification accuracy between FedCPD and baselines on six datasets.

Algorithm	Dermoscopic(4)	Office(4)	Digit5(4)	DomainNet(5)	CIFAR10(4)	FMNIST(4)
FedAvg	69.31	62.82	83.69	49.21	32.84	67.08
FedProx	68.54	62.87	83.87	49.36	36.79	71.58
FedDF	70.95	*	84.39	49.27	36.46	67.68
FedDisco	73.71	66.05	84.68	50.11	36.80	66.98
FedSplit	71.53	62.76	83.20	48.75	33.84	67.64
FedGen	70.57	62.53	82.79	47.71	33.56	69.96
FedCG	72.96	67.42	84.82	49.80	35.78	70.94
FedCPD	**74.32**	**71.29**	**85.28**	**52.36**	**39.14**	**75.48**

TABLE 4. Comparison of classification accuracy of FedCPD under different pseudo dataset construction algorithms.

Algorithm	Dermoscopic(4)	Office(4)	Digit5(4)	DomainNet(5)	CIFAR10(4)	FMNIST(4)
LDM	**74.32**	71.29	**85.28**	**52.36**	39.14	**75.48**
GDM(T=1)	74.19	**71.53**	85.01	51.83	38.68	75.20
GDM(T=2)	73.89	71.47	84.98	51.76	**39.23**	74.96
GDM(T=3)	74.08	71.52	85.17	52.20	38.75	75.28

Throughout all experiments, the hyperparameters w and v in the global model weight aggregation are set to 0.3 and 0.5, respectively. For the two non-negative hyperparameters balancing the loss functions, μ is initially set to 0 and λ is set to 0.5. μ gradually increases from 0 to 1 based on the number of rounds already trained locally. For the non-IID setting, the dirichlet distribution parameter α is set to 0.5. The default method for constructing pseudo dataset is LDM. The calculation of LCD employs the L2 norm.

3) BASELINES

We select seven baseline methods to compare with our framework. These baselines contain two types, one that shares all client model parameters with the server, such as FedAvg [37], FedProx [32], FedDF [33], and FedDisco [10]. Among them, FedAvg is a single-model FL approach that obtains a global model by weighted averaging the local models. FedProx introduces a proximal term in the local objective to standardize the local model training. FedDF uses knowledge distillation to aggregate local models on the server using unlabeled public data. FedDisco, on the other hand, improves on FedAvg by introducing local differences in the aggregation weights.

The other is to share only some of the model parameters with the server, and we pick FedSplit [34], FedGen [35], and FedCG [36] as the baseline methods. FedSplit shares only the public classifiers of the local network to protect privacy. FedGen uses knowledge distillation to train the global generator, which in turn helps clients train their local networks. FedCG uploads only the public classifiers and uses knowledge distillation for server-side aggregation.

B. EXPERIMENT RESULTS

1) PERFORMANCE EVALUATION ON SIX DATASETS

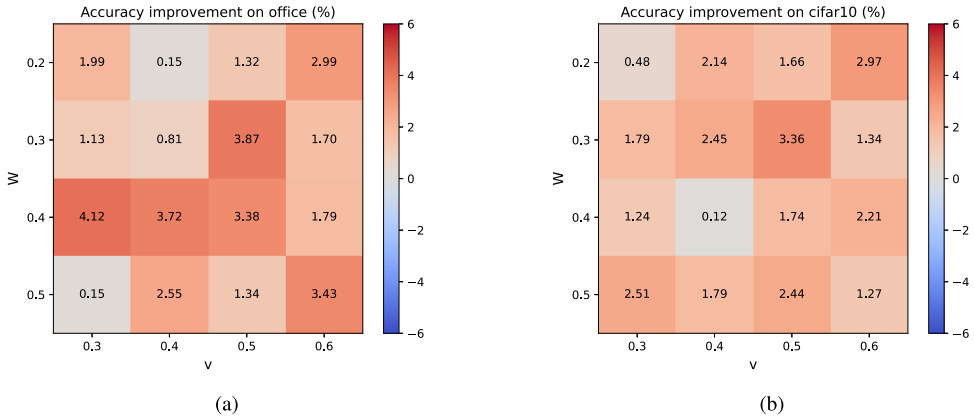
Table 3 shows a comparison of the classification accuracies of FedCPD and other FL baseline methods on all six datasets. The first four datasets contain data from multiple domains, so they naturally conform to the non-independent homogeneous distribution setting. For Cifar10 and Fmnist, we partitioned them via the Dirichlet distribution.

On all six datasets, FedCPD achieves the highest classification accuracy. The experiments demonstrate that our proposed FedCPD outperforms other baseline methods when dealing with non-IID datasets, showing its effectiveness. In contrast, FedAvg exhibits poorer performance on almost all datasets, indicating that the global model based on weighted average cannot effectively handle non-IID data.

FedCPD utilizes DBAW and a globally shared pseudo dataset to address the impact of local data imbalance across clients on model training. As a result, FedCPD has a greater advantage over other baseline methods in dealing with non-IID data. For the first four datasets, on Office, FedCPD improves the classification accuracy by at least 3.87% compared to other baseline methods. Whereas, on Digit5, FedCPD has a relatively small advantage in classification accuracy over the other baseline methods. This is because Office has more severe heterogeneity and imbalance compared to Digit5. For Cifar10 and Fmnist, which are partitioned using the Dirichlet distribution, FedCPD improves by 2.34% and 3.9%, respectively, over the suboptimal methods. For the baseline methods, the imbalance of the datasets leads to poorer classification results for unknown data by the classifiers of its local model. In addition, the baseline methods use the dataset size as the aggregation weight. This leads to the fact that client models with higher performance but less data size cannot be assigned larger aggregation weights, which seriously affects the performance of the aggregated model.

2) IMPACT OF DIFFERENT METHODS FOR GENERATING PSEUDO DATASET

Table 4 shows the performance of FedCPD on different types of pseudo dataset. The experimental results show



(a) (b)

FIGURE 3. Classification accuracy of FedCPD on Office and Cifar10 under different values of hyperparameters *w* and *v*. (a) Office. (b) Cifar10.

TABLE 5. Comparison of classification accuracy of FedCPD at different discrepancy measurement scales.

METRIC	Dermoscopic(4)	Office(4)	Digit5(4)	DomainNet(5)	CIFAR10(4)	FMNIST(4)
COSINE	74.15	70.82	84.91	52.29	39.05	75.42
KL-DIVERGENCE	74.22	71.14	**85.39**	**52.44**	**39.23**	75.20
L1 Norm	74.28	71.23	85.12	52.37	38.76	75.34
L2 Norm	**74.32**	**71.29**	85.28	52.36	39.14	**75.48**

that FedCPD outperforms other baseline methods regardless of the pseudo dataset construction method adopted. From Table 2, we are also able to see that FedCPD achieves the highest accuracy on four datasets using LDM. It uses GDM on the remaining two datasets and achieves the highest accuracy at k = 1 and k = 2, respectively. This shows that both pseudo dataset construction methods are effective. And by using GDM, it is possible to achieve comparable classification accuracy to LDM while protecting user data privacy.

3) IMPACT OF DIFFERENT HYPERPARAMETER VALUES

Fig. 3 shows the impact of different values of hyperparameters w and v on classification accuracy on both Office and Cifar10. The numbers in the figure indicate the change in classification accuracy of FedCPD compared to FedCG for different values of w and v. The experiments demonstrate that FedCPD consistently improves performance when w is between 0.2 and 0.5 and v is between 0.3 and 0.6. Specifically, for Office, FedCPD shows relatively stable performance improvement when w is between 0.3 and 0.4. For Cifar10, FedCPD exhibits superior and stable performance in classification accuracy when v is between 0.5 and 0.6. Therefore, choosing w=0.3 and v=0.5 is undoubtedly a better option. This experiment also proves the effectiveness and robustness of the aggregation weights based on discrepancies.

4) IMPACT OF DIFFERENT DISCREPANCY METRICS

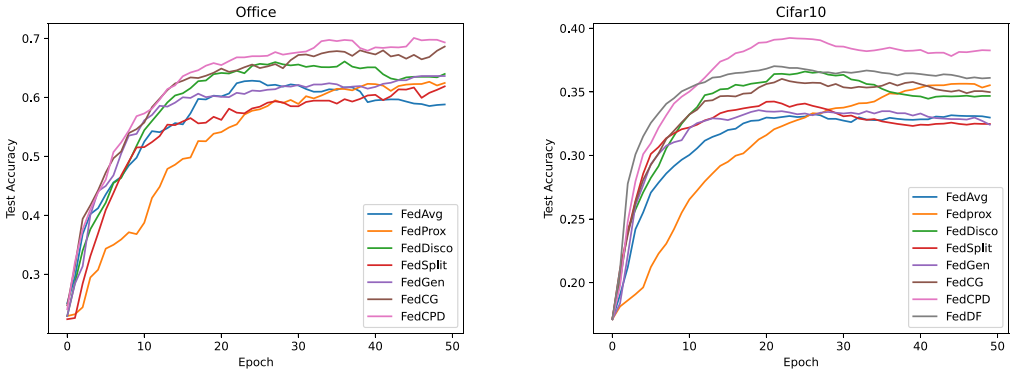
In our experiments, we default to using L2 norm to compute local discrepancies. Table 5 shows the impact of local discrepancies on classification accuracy under different measurement scales. We can see that across all six datasets, using L2 norm to compute local discrepancies achieves the highest classification accuracy on Dermoscopic, Office, and Fmnist. On the other hand, using KL divergence to compute local differences achieves the best performance on the remaining three datasets. Table 5 indicates that FedCPD achieves similar performance across different local difference measurement scales. This demonstrates the robustness of FedCPD to different difference measurement scales and its ability to effectively utilize local differences for more efficient weight aggregation.

5) CONVERGENCE ANALYSIS

To validate the convergence of FedCPD, we conducted experiments on Office and Cifar10 with 100 and 50 iterations respectively for convergence analysis. From Fig. 4, it can be observed that FedCPD consistently outperforms all FL baseline methods. In terms of convergence speed, FedCPD exhibits comparable performance on both datasets. FedAvg and FedProx show lower accuracy and slower convergence speed on Office and Cifar10, likely due to the negative transfer effects caused by data imbalances.

6) ABLATION STUDY

To further demonstrate the effectiveness of global shared pseudo dataset and DBAW, we conduct a simple ablation experiment. FedCPD is based on FedCG, which corrects classifiers using a globally shared pseudo dataset and calculates aggregation weights based on discrepancies. Therefore, we denote FedCG with a globally shared pseudo dataset as FedCG(p), and FedCG with DBAW as FedCG(d). We conducted experiments on six datasets using FedCG, FedCG(p), FedCG(d), and FedCPD, respectively.



(a) (b)

FIGURE 4. Convergence curves of FedCPD and baselines on Office and Cifar10. (a) Office. (b) Cifar10.

TABLE 6. Impact of DBAW and pseudo dataset on classification accuracy.

Algorithm	Dermoscopic(4)	Office(4)	Digit5(4)	DomainNet(5)	CIFAR10(4)	FMNIST(4)
FEDCG	72.96	67.42	84.82	49.80	35.78	70.94
FEDCG(p)	73.83	70.09	84.78	51.90	37.56	71.27
FEDCG(d)	73.28	69.98	85.06	50.76	38.23	72.69
FEDCPD	**74.32**	**71.29**	**85.28**	**52.36**	**39.14**	**75.48**

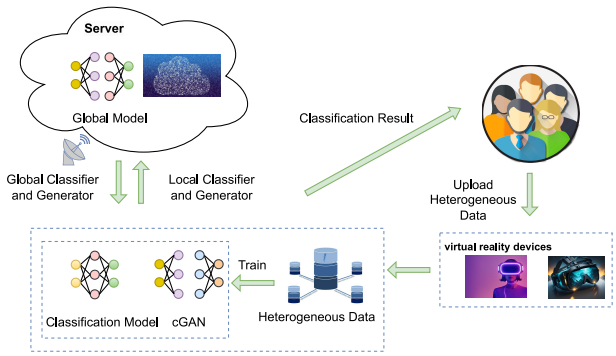


FIGURE 5. Application of FedCPD in the metaverse.

As shown in Table 6, the experiments demonstrate that both global shared pseudo dataset and DBAW effectively improve classification accuracy. FedCPD achieves the highest accuracy among the four methods on all six datasets. Particularly, on Office, both FedCG(p) and FedCG(d) show at least a 2.56% improvement in classification accuracy compared to FedCG. However, on Digit5, FedCG(p) is the only method that leads to a performance drop, with a decrease of 0.04% in classification accuracy. This may be due to the small inherent differences in Digit5, where the introduced pseudo dataset incorrectly influences the classifier.

By comparing the classification accuracies of the four methods, we can draw the following conclusions: 1) Using DBAW significantly improves the model's classification accuracy compared to traditional aggregation weights based on dataset sizes; 2) Utilizing synthetic global shared pseudo dataset can correct local classifiers, but it may lead to performance degradation on datasets with mild imbalances and heterogeneity.

C. APPLICATION OF FEDCPD IN THE METAVERSE

In the metaverse, a plethora of virtual reality devices such as head-mounted displays and full-body tracking systems are prevalent. Integration of FedCPD into these virtual reality devices facilitates the acquisition of precise classification outcomes while ensuring user data security. As depicted in Fig. 5, FedCPD tailors local classification models for individual metaverse users by leveraging locally distributed data across virtual reality devices. Throughout the training process, the user's local data remains confined to their respective devices, thereby safeguarding data integrity. Furthermore, due to the heterogeneity inherent in the data collected from diverse virtual reality devices used by metaverse users, the performance of classification models may be compromised. FedCPD addresses this challenge by employing DBAW and a globally shared, label-independent pseudo dataset to enhance the performance of both global and local models, thus aligning classification outcomes more closely with local data characteristics.

In comparison to previous methods, the incorporation of FedCPD offers metaverse users enhanced security in their participation and yields more precise classification outcomes.

V. CONCLUSION

In this paper, we proposed FedCPD, an FL algorithm to secure and process heterogeneous data in the metaverse. It used a cGAN-based privacy-preserving method to secure user data. Specifically, FedCPD utilized generators in cGAN

instead of extractors to participate in server-side aggregation to avoid the exposure of feature extractors to the server. Moreover, FedCPD employed a novel aggregation mechanism that assigned greater aggregation weights to better-performing local models by utilizing DBAW to perform aggregation operations. It mitigated the impact of data heterogeneity on the global model. Finally, we designed a correction module in FedCPD to correct the classifier with a globally shared, label-independent pseudo dataset. It improved the classifier's ability to classify unknown data in the metaverse. Extensive experiments across various datasets indicated that FedCPD achieved the highest classification accuracy, showcasing its effectiveness in addressing heterogeneous data. Therefore, FedCPD could be applied in the metaverse to achieve secure and accurate image classification. In the future, we plan to enhance FedCPD by integrating more effective personalized FL methods and novel privacy protection technologies. Given that FL is highly susceptible to attacks during the communication process, we will focus on improving FedCG's privacy protection capabilities in communication, paving the way for more advanced FL applications in the metaverse.

REFERENCES

[1] M. Zawish et al., "AI and 6G into the metaverse: Fundamentals, challenges and future research trends," *IEEE Open J. Commun. Soc.*, vol. 5, pp. 730–778, 2024.

[2] Y. Huang, Y. J. Li, and Z. Cai, "Security and privacy in metaverse: A comprehensive survey," *Big Data Min. Anal.*, vol. 6, no. 2, pp. 234–247, 2023.

[3] T. Zhang, L. Gao, C. He, M. Zhang, B. Krishnamachari, and A. S. Avestimehr, "Federated learning for the Internet of Things: Applications, challenges, and opportunities," *IEEE Internet Things Mag.*, vol. 5, no. 1, pp. 24–29, Mar. 2022.

[4] M. A. Serhani, H. G. Abreha, A. Tariq, M. Hayajneh, Y. Xu, and K. Hayawi, "Dynamic data sample selection and scheduling in edge federated learning," *IEEE Open J. Commun. Soc.*, vol. 4, pp. 2133–2149, 2023.

[5] Y. Chen, S. Huang, W. Gan, G. Huang, and Y. Wu, "Federated learning for metaverse: A survey," in *Proc. ACM Web Conf. Compan.*, 2023, pp. 1151–1160.

[6] H. Chen, T. Zhu, T. Zhang, W. Zhou, and P. S. Yu, "Privacy and fairness in federated learning: On the perspective of tradeoff," *ACM Comput. Surveys*, vol. 56, no. 2, pp. 1–37, 2023.

[7] H.-Y. Tran, J. Hu, X. Yin, and H. R. Pota, "An efficient privacy-enhancing cross-silo federated learning and applications for false data injection attack detection in smart grids," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 2538–2552, 2023.

[8] S. Zeng et al., "HFedMS: Heterogeneous federated learning with memorable data semantics in industrial metaverse," *IEEE Trans. Cloud Comput.*, vol. 11, no. 3, pp. 3055–3069, Sep. 2023.

[9] M. H. Mahmoud, A. Albaseer, M. Abdallah, and N. Al-Dhahir, "Federated learning resource optimization and client selection for total energy minimization under outage, latency, and bandwidth constraints with partial or no CSI," *IEEE Open J. Commun. Soc.*, vol. 4, pp. 936–953, 2023.

[10] R. Ye, M. Xu, J. Wang, C. Xu, S. Chen, and Y. Wang, "Feddisco: Federated learning with discrepancy-aware collaboration," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 39879–39902.

[11] Y. Guo, X. Tang, and T. Lin, "FedBR: Improving federated learning on heterogeneous data via local learning bias reduction," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 12034–12054.

[12] 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," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 5319–5329.

[13] L. Belcastro, F. Marozzo, A. Orsino, D. Talia, and P. Trunfio, "Edge-cloud continuum solutions for urban mobility prediction and planning," *IEEE Access*, vol. 11, pp. 38864–38874, 2023.

[14] G. Thakur, P. Kumar, C.-M. Chen, A. V. Vasilakos, Anchna, and S. Prajapat, "A robust privacy-preserving ECC-based three-factor authentication scheme for metaverse environment," *Comput. Commun.*, vol. 211, pp. 271–285, Nov. 2023.

[15] K. Yang, Z. Zhang, Y. Tian, and J. Ma, "A secure authentication framework to guarantee the traceability of avatars in metaverse," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 3817–3832, 2023.

[16] C. Li, L. Zeng, X. Huang, X. Miao, and S. Wang, "Secure semantic communication model for black-box attack challenge under metaverse," *IEEE Wireless Commun.*, vol. 30, no. 4, pp. 56–62, Aug. 2023.

[17] L. Sun, C. Li, B. Liu, and Y. Zhang, "Class-driven graph attention network for multi-label time series classification in mobile health digital twins," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 10, pp. 3267–3278, Oct. 2023.

[18] X. Zhou et al., "Personalized federation learning with model-contrastive learning for multi-modal user modeling in human-centric metaverse," *IEEE Sensors J. Sel. Areas Commun.*, vol. 42, no. 4, pp. 817–831, Apr. 2024.

[19] J. Chen, J. Wang, C. Jiang, Y. Ren, and L. Hanzo, "Trustworthy semantic communications for the metaverse relying on federated learning," *IEEE Wireless Commun.*, vol. 30, no. 4, pp. 18–25, Aug. 2023.

[20] K. Bonawitz et al., "Practical secure aggregation for privacy-preserving machine learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 1175–1191.

[21] N. C. Codella et al., "Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (ISBI), hosted by the international skin imaging collaboration (ISIC)," in *Proc. IEEE 15th Int. Symp. Biomed. Imag. (ISBI 2018)*, 2018, pp. 168–172.

[22] P. Tschandl, C. Rosendahl, and H. Kittler, "The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions," *Sci. Data*, vol. 5, no. 1, pp. 1–9, 2018.

[23] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2066–2073.

[24] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, "Moment matching for multi-source domain adaptation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1406–1415.

[25] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proc. 11th Eur. Conf. Comput. Vis., (ECCV)*, Crete, Greece, 2010, pp. 213–226.

[26] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," Dataset, California Inst. Technol., Pasadena, CA, USA, 2007.

[27] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[28] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by back-propagation," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1180–1189.

[29] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *Proc. NIPS Workshop Deep Learn. Unsupervised Feature Learn.*, 2011, pp. 1–9.

[30] J. J. Hull, "A database for handwritten text recognition research," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 5, pp. 550–554, May 1994.

[31] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*.

[32] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst.*, 2020, pp. 429–450.

[33] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 2351–2363.

[34] H. Gu, L. Fan, B. Li, Y. Kang, Y. Yao, and Q. Yang, "Federated deep learning with Bayesian privacy," 2021, *arXiv:2109.13012*.

[35] Z. Zhu, J. Hong, and J. Zhou, "Data-free knowledge distillation for heterogeneous federated learning," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 12878–12889.

[36] Y. Wu, Y. Kang, J. Luo, Y. He, and Q. Yang, "FedCG: Leverage conditional GAN for protecting privacy and maintaining competitive performance in federated learning," 2021, *arXiv:2111.08211*.
[37] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artif. Intell. Statist.*, 2017, pp. 1273–1282.
[38] S. Le, W. Yueyuan, R. Yongjun, and X. Feng, "Path Signature-based XAI-enabled network time series classification," *Sci. China Inf. Sci.*, vol. 67, Jun. 2024, Art. no. 170305.
[39] C. Chen, Y.-H. Chiang, H. Lin, J. C. Lui, and Y. Ji, "Joint client selection and receive beamforming for over-the-air federated learning with energy harvesting," *IEEE Open J. Commun. Soc.*, vol. 4, pp. 1127–1140, 2023.
[40] L. Wulfert et al., "AifES: A next-generation edge AI framework," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 6, pp. 4519–4533, Jun. 2024.

LE SUN received the Ph.D. degree from Victoria University, Australia, in 2016. She is a Professor with the School of Computer, Nanjing University of Information Science and Technology, China. She was a Visiting Scholar with Michigan State University in 2018. She has published over 50 papers in high-quality conferences and journals. Her research interests include intelligent medicine and services computing. She received the Chancellor-Citation Best Ph.D. Award.

ZHIMENG ZHANG received the bachelor's degree from the Nanjing University of Information Science and Technology, China, in 2022, where he is currently pursuing the master's degree with the School of Computer. His research interests include deep learning and federated learning.

GHULAM MUHAMMAD (Senior Member, IEEE) received the B.S. degree in computer science and engineering from the Bangladesh University of Engineering and Technology in 1997, and the M.S. and Ph.D. degrees in electronic and information engineering from Toyohashi University and Technology, Japan, in 2003 and 2006, respectively. He is a Professor with the Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia. He has authored and co-authored more than 300 publications including IEEE / ACM / Springer / Elsevier journals, and flagship conference papers. He owns two U.S. Patents. He is involved in many research projects as a principal investigator and a co-principal investigator. His research interests include signal processing, machine learning, IoTs, medical signal and image analysis, AI, and biometrics. He was a recipient of the Japan Society for Promotion and Science Fellowship from the Ministry of Education, Culture, Sports, Science and Technology, Japan.