International Journal of Engineering Trends and Technology
ISSN: 2231-5381/ https://doi.org/10.14445/22315381/IJETT-V74I11P104

Volume 74 Issue 11, 51-66, November 2026
© 2026 Seventh Sense Research Group®

Original Article

Clustering-based Personalized Deep Federated Learning using Cluster-Conditional Federated Generative Adversarial Networks

Nandakumar Mani[1*], Ranjith Nadarajan[1]

[1]*Department of Computer Science, K.S.G College of Arts and Science, Tamilnadu, India.*

[*]*Corresponding Author: nandakumarmphd9891@gmail.com*

Received: 29 April 2026 Revised: 12 September 2026 Accepted: 18 September 2026 Published: 25 November 2026

Abstract *- The fast development of Internet of Things (IoT) devices has resulted in massive volumes of decentralized data, simultaneously raising challenges related to data heterogeneity, communication overhead and privacy leakage. Conventional Machine Learning (ML) approaches gather data from IoT devices to a cloud server to train a model, causing high communication costs, delays and a lack of privacy, because they expose data to the server. Federated Learning (FL), by contrast, addresses privacy issues by keeping data local on devices. However, standard FL still suffers from non-identical data distributions, limited personalization, scalability issues and poor convergence in heterogeneous IoT environments. To solve the above issues, this paper proposes a Privacy-Preserving Cluster Conditional Federated Generative Adversarial Network supported Clustering-based Personalized Deep FL (PPCFGAN-CPDFedL) for heterogeneous Cloud–IoT environments. The proposed model integrates PPCFGAN, Conditional Wasserstein GAN with Gradient Penalty (WGAN-GP), and Differential Privacy (DP) to generate safe, stable and private data locally on the client side. The proposed classifier is a Capsule-Long Short-Term Memory with Self-Attention (CLSTM-SA) model to improve classification and prediction accuracy. On the server side, a pre-trained autoencoder computes latent client embeddings, enabling similarity-based clustering of client models and weighted collaborative aggregation. Experimental results on the E-Commerce and CIFAR-100 datasets show that the new PPCFGAN-CPDFedL has the best performance with 93.26% and 94.13% accuracy under the challenging Group Non-IID and Client Non-IID scenarios, compared to the existing models, including PGFL, PDP-FL, DP-CFL and pFedCAM models in all the metrics.*

Keywords *- Cloud-IoT, Data heterogeneity, Federated Learning, Privacy-Preserving GAN.*

1. Introduction

A new age of the Internet of Things (IoT) has begun with the fast expansion of computing and smart devices, which has allowed for enormous advancements in many fields, including smart healthcare systems, intelligent transportation, and industrial automation [1]. The use of IoT devices produces large amounts of user data that can be used for extracting information and developing ML models to deliver quality intelligent services.

Intelligent IoT applications conventionally rely on aggregating decentralized user data onto a central cloud server to train machine learning models. After training completes, the global model is deployed back to edge devices to perform local inference.

However, the continuous data transfers and model updating of these approaches incur high latency and communication costs [2]. There are a lot of privacy and security issues that come up when users share sensitive information with cloud servers. General Data Protection Regulation (GDPR) [3] further ensure the security of data by limiting the transferring data to the centralized nodes.

A distributed learning mechanism was proposed to train and update models directly on IoT devices using local data, thus avoiding data transmission. However, this distributed learning approach faces certain challenges, such as the computational, energy and storage requirements for local learning. Furthermore, the relatively small sample size of local data and uneven data distributions at individual IoT devices can result in poor convergence of models and inferior performance.

FL is an innovative strategy that solves the difficulties of IoT training with distributed data. FL enables collaborative training of a global model by aggregating and averaging local updates from IoT devices [4]. The primary advantage of this approach is decoupling of training models from the direct use of data. FL safeguards both data privacy and security by

This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

performing calculations on IoT devices themselves while only exchanging model parameters or gradients. In turn, this allows devices to learn a high-performing global model that preserves privacy-sensitive user data and remains efficient while adhering to privacy protection laws.

In traditional FL, it is not possible to have a consensus on a common model for cooperative learning among all the devices that participate in the application. But, for accurate and complex Cloud-IoT applications, consensus needs to be achieved. To address these diversity issues, a Personalized FL (PFL) approach is proposed, which personalizes at device, data and model levels. This personalization takes care of the non-identical data distributions, computation capabilities and model drift across devices. This way, a good quality model is obtained for the IoT devices to make accurate inferences. With the flexibility, recently, many researchers have used personalized training. PFL model is used in IoT based personal healthcare systems [5], smart home automations [6] and location aware recommendation systems [7] and real-time smart video analysis [8].

Personalized FL (PFL) enhances learning capacity of the model by learning from the personalized data of each client, but there are drawbacks in large-scale and diverse IoT settings. Training separate models for each individual device would be inefficient and would lead to scalability issues, potentially leading to underutilization of resources and the computational burden, particularly in the case of multiple clients with similar characteristics in data [9]. To solve such problems, the Clustering-based FL (CFL) has been proposed. CFL overcomes this by clustering clients based on certain criteria and then building a separate model for each cluster. This signifies that users with comparable data distributions can be aggregated, lessening the influence of data variability on model performance.

Iterative CFL (ICFL) models effectively handle client heterogeneity by reclustering clients. For instance, the ICFL model proposed in [10] groups clients together according to the cosine similarity of gradient at each iteration, while ICFL model in [11] assigns nodes to a predetermined number of clusters by minimizing losses through loss functions. These repetitive partitioning schemes provide best performance over the standard FL but have high communication overhead and computations at edge nodes. Static (non-iterative) clustering methods, on the other hand, do not incur this overhead and are inflexible, as they cannot fix misaligned clusters at a later training stage.

The limitations of the PFL and CFL models highlighted the need to have a flexible and scalable solution that could incorporate the best qualities of both the clustering approach and the personalization approach. pFedCK [12] integrates clustering and local personalization with knowledge distillation. In this model, each client executes a dual-model system: one for the global aggregation and one for the local execution. The server clusters the nodes according to the resemblance in the degree of parameter fluctuation submitted by various interaction models in each training iteration.

Nevertheless, sharing local parameter updates or gradients may still lead to privacy breaches. DP [13] is adapted at the client level and sample level by gradient pruning to reduce the sensitivity of the global gradient and adding random noise. Noise injection ensures the privacy of the data, but can affect the utility of the model and the speed of training. To address this problem of privacy and utility, DP-based FL Algorithm with Clustered Model Random Selection (DPFedCMRS) [14] was proposed, which comprises an adaptive personalized clustering mechanism with a randomized cluster selection scheme, hence converging quickly on heterogeneous data without sacrificing privacy guarantees.

1.1. Problem Statement

Conventional Differential Privacy in FL (DP-FL) requires fixed noise scales and uniform clipping thresholds for all the nodes involved in the FL. This rigid design neglects the fact that there are different kinds of data at the client level, resulting in a strong mismatch between privacy and utility. Too much noise leads to a decrease in the accuracy and convergence speed of the model, and too little noise makes the parameters of the client machine vulnerable to model inversion attacks. Thus, applying the same DP parameters to non-IID client distributions is not optimal in terms of achieving a balance between privacy and utility, and adaptive (client-specific) privacy techniques are needed.

1.2. Major Contributions

The PPCFGAN-CPDFedL model enhances personalized and privacy-preserving FL in the context of the heterogeneous Cloud-IoT platforms. The main aim of the model is to overcome critical problems like data imbalance, client heterogeneity and privacy leakage. The whole process of the proposed approach is as follows:

1.2.1. Client Side (Edge Layer)

- Each client manages heterogeneous edge devices (e.g., smart sensors, voice assistants, mobile hardware) generating local, non-IID data streams. To perform local classification and prediction tasks, clients employ a hybrid CLSTM-SA classifier. Within this architecture, Capsule Networks (CapsNets) replace traditional Convolutional Neural Network (CNN) layers to preserve hierarchical spatial relationships within localized data streams. Additionally, a Self-Attention (SA) convolution module captures long-range contextual correlations while filtering out irrelevant environmental noise.
- Concurrently, each client trains a local Conditional PPCFGAN model using a Wasserstein GAN with Gradient Penalty (WGAN-GP) backbone to stabilize

generative training. The local PPCFGAN models and augment localized feature distributions to address local class imbalances. To guarantee strong user privacy, DP is integrated directly into the PPCFGAN training process by applying gradient clipping and injecting calibrated noise.

- Upon completing local updates, edge devices transmit only their trained local PPCFGAN parameters (Generator G and Discriminator D) to the cloud server. Raw private data is never transmitted, significantly reducing bandwidth overhead while shielding local data from exposure.

1.2.2. Server Side (Cloud layer)

- Upon receiving the uploaded local PPCFGAN parameters, the cloud server combines the generative models ($G + D$) to synthesize a comprehensive, balanced Synthetic IID Dataset that captures the global feature distribution across all participating clients.
- The generated synthetic dataset is passed through a pretrained Autoencoder (AE) to extract low-dimensional latent embeddings for each client profile. The server then computes pairwise representation distances to construct an inter-client Similarity Matrix.
- Utilizing the similarity matrix, the central server groups participating nodes into distinct, homogeneous clusters (Cluster 1 through Cluster n), placing clients with similar data distributions into shared groups.
- Then redistributed back to the clients, where it can then be used for personalized and federated model updates.
- Within each cluster, the server performs a weighted collaborative aggregation, granting higher contribution weights to clients with higher representation similarities. A new, cluster-specific global PPCFGAN model is trained on the aggregated cluster parameters and redistributed back to the corresponding client groups to drive personalized local model updates.

The later sections will explain each technique in detail.

2. Literature Survey

Gauthier et al. [15] proposed a Personalised Graph FL (PGFL) framework that facilitates the collaborative development of cluster-specific models while maintaining data confidentiality. The method leverages graph-based relationships among clients to exploit inter-cluster similarities and integrates zero-Concentrated DP (zCDP) by perturbing model exchanges with noise. However, very large values of learning parameters can lead to performance degradation.

Liu et al. [16] proposed a privacy budget-agnostic Personalized Differential Privacy (PDP) framework to improve the privacy of FL models. This approach does not require the disclosure of privacy parameters. Rather, it clusters clients in a privacy-homogeneous grouping of clients indirectly by their noisy model updates. To separate the effects of DP noise and non-IID data, the paper carefully investigates the situation where model updates are mainly affected by the level of noise added to the updates rather than their data distribution. Leveraging this idea, a method was recommended for grouping based on the L_2-norm of the noisy model updates. However, this approach could be less effective when the privacy budget allocated to the clients is equal or hard to rank.

Yang et al. [17] proposed a pFedCAM algorithm that enhances the robustness of FL algorithms to device and data heterogeneity. This algorithm incorporates client clustering and model interpolation. First, the heterogeneous clients are grouped into clusters, and parallel FedAvg is performed within clusters. Then, models from different clusters are further interpolated to create individual global models. However, multi-level clustering and interpolation algorithm incurs significant computational costs.

Malekmohammadi et al. [18] proposed a DP-CFL method to address data heterogeneity and privacy issues. It uses DP and client-level clustering to improve the reliability of federated systems. In particular, clients are grouped using model updates as well as training loss. To counteract the effects of clustering errors and server uncertainty, the method uses a large batch size and Gaussian Mixture Models (GMM), reducing the impact of stochastic and DP noise in the model aggregation process. But this may suffer from poor performance in cases of extremely high data diversity and skewed client distributions.

To improve privacy and model performance, Wang et al. [19] introduced Social-aware Clustered FL (SCFL), which exploits users' social network information. SCFL groups mutually trusted users to form social clusters and aggregating their raw model updates before feeding them into the cloud to prevent malicious users from obtaining private data. But this approach adds extra computational and communication costs for social cluster creation.

Pan et al. [20] proposed a personalized load-forecasting framework for residential energy users that addresses data heterogeneity and privacy concerns in smart grids. The framework enables users to train personalized forecasting models without sharing raw load data.

First, user load pattern similarities are computed under privacy-preserving mechanisms. Next, a network is established, and a federated user grouping technique utilizes the Louvain approach to combine users into groups with similar load patterns.

Finally, a customized and flexible DP-FL-LSTM model is employed for load forecasting. However, the prediction performance of the model may vary across clusters, resulting in uneven accuracy for different groups of users.

Wei et al. [21] devised a secure and communication-efficient CFL framework integrating DP to handle heterogeneous Base Stations (BSs) and non-IID user data. The study formulates a joint optimization problem for resource block allocation, user scheduling, and DP noise control to minimize training loss and transmission delay. To solve this, a DP-assisted Value Decomposed Multi-Agent Reinforcement Learning (DPVD-MARL) technique was developed, which collaboratively optimize learning performance while preserving privacy. However, the framework may be less effective in networks with extremely high user heterogeneity or communication delays.

2.1. Research Gaps

The existing CFL and PFL approaches are subject to several drawbacks, such as uniform DP settings that do not reflect client heterogeneity, fixed clustering methods to achieve flexibility, and limited personalization in non-IID data setups. Many networks are also vulnerable to privacy breaches through gradient sharing, and some are also unstable in the creation of synthetic data with GAN.

Further, the conventional definitions of similarity do not consider the actual client associations, and this leads to a sub-optimal aggregation strategy. The persistent issues highlight that the current approaches are yet to deliver the best solutions, and hence, more work is needed to solve the problems.

3. Proposed Methodology

The PPCFGAN-CPDFedL model is detailed in this section. The system consists of IoT devices and edge clients that produce and store data, and a cloud server that oversees the clustering, aggregation and the update of global model. The diagrams of the proposed model are in Figure 1.

3.1. Privacy Preserving Cluster Conditional Federated Generative Adversarial Network (PPCFGAN)

The G in the GAN is responsible for generating samples similar to real samples, whereas the D would try to discriminate created samples from real samples. While training, D and G play minimax by continuously adjusting their parameters until they reach a stable state. The loss function of GAN is described as:

$$\mathcal{L}_{GAN} = \mathbb{E}_{x\sim P_{data}(x)}\left[\log\left(D(x)\right)\right] + \mathbb{E}_{z\sim P_z(z)}\left[\log\left(1 - D\left(G(z)\right)\right)\right] \quad (1)$$

Where P_x, P_z denote the real and synthetic data distribution.

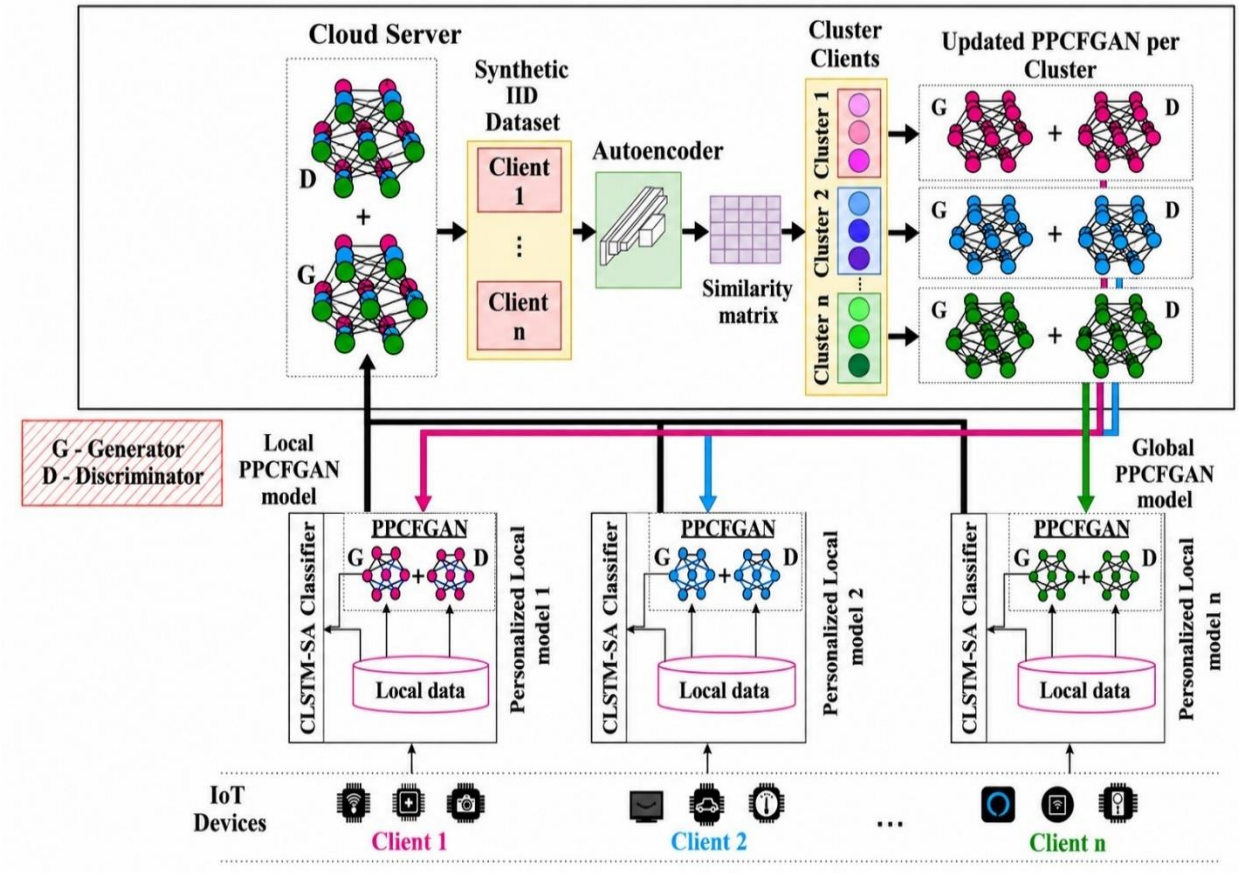

Fig. 1 Schematic representation of the proposed model

However, normal GANs continue to have issues with unstable training, disappearing gradients, and mode collapse. The solution to these issues is derived based on WGAN-GP.

The WGAN is a standout improvement to the GAN. It uses a different objective function (Wasserstein distance) than a normal GAN loss function, and a gradient norm penalty, which enforces the Lipschitz constraint to further improve training stability.

Given the actual data x, the input noise $z \sim P_z$ and the synthetic data $G(z)$, the gradient penalty term can be written as:

$$(\|\nabla_{\hat{x}} D(\hat{x})\| - 1)^2 \quad (2)$$

Where $\hat{x}$ is a weighted average of the genuine and fake data. The loss function for G and D is formulated as:

$$\mathcal{L}_G = D\big(G(z)\big) \quad (3)$$

$$\mathcal{L}_D = D(x) - D\big(G(x)\big) + \lambda(\|\nabla_{\hat{x}} D(\hat{x})\| - 1)^2 \quad (4)$$

Where λ represents the weight of the gradient penalty.

Here, P_z is set to follow a standard Gaussian distribution $N(0,1)$ and $\lambda = 10$ with gradient penalty. Similarly, to Equation (6), the loss function of WGAN may be expressed as:

$$\mathcal{L}_{WGAN} = \mathbb{E}_{G(z)\sim P_z}\big[D\big(G(z)\big)\big] - \mathbb{E}_{x\sim P_x}[D(x)] + \lambda \mathbb{E}_{\hat{x}\sim P_{G(z)}}\left[(\|\nabla_{\hat{x}} D(\hat{x})\| - 1)^2\right] \quad (5)$$

While WGAN-GP ensures stable training and realistic sample generation, it does not inherently protect data privacy. To solve this, DP is used to prevent the model from revealing critical information about the training data by introducing intentionally random noise to the gradients. This method ensures that the core data remains private while letting the G learn realistic data distributions.

DP is a stochastic statistical technique represented as $\mathcal{M}(\cdot)$ that complicates the observer's ability to discern the input data. An observer refers to any individual seeking to derive insights from the algorithm's results regarding the data collection. The current definition of DP is as follows.

Definition 1. For every two neighbouring datasets D and D', and for every randomly generated dataset S, a randomised method $\mathcal{M}$ is considered to satisfy DP if and only if the following condition is true:

$$Pr(\mathcal{M}(D) \in S) \leq e^{\varepsilon}.Pr(\mathcal{M}(D') \in S) + \delta \quad (6)$$

Where $Pr(\cdot)$ denotes the potential outcome using the randomised method. ε is the privacy budget controlling the level of privacy, and δ represents the likelihood of not necessarily fulfilling distinct confidentiality. DP technique protects client data during synthetic samples generation, while retaining useful representations for feature learning. In the next stage, each client then employs the CLSTM-SA classifier to perform accurate data classification and prediction tasks locally.

3.2. CLSTM-SA for Classification and Prediction

3.2.1. Capsule Networks and Long Short-Term Memory (CLSTM)

The traditional CNN algorithms make predictions by identifying the presence of features in an image. But CNNs are unable to detect spatial patterns between features, and these require a large amount of data to train and perform well. The pooling operation used in CNNs to reduce dimensionality and computational load was identified as a major factor contributing to limited functionality of CNN. To address these limitations, Hinton [6] proposed the CapsNet architecture. It is a type of neural network where the fundamental unit is called as capsule. Each capsule receives input and produces output in the form of vectors. Let the input vectors from the lower capsule layer be u_i, which are multiplied by corresponding weight matrices W_{ij} and coupling factors c_{ij}, with i, j denotes the capsule index in the lesser and higher layers, as defined in equation:

$$s_j = \sum_{i=1}^{N} c_{ij} W_{ij} u_i \quad (7)$$

The resulting vector s_j is aggregated element-wise, then normalized via a squashing function to produce the outcome vector v_j, is expressed as:

$$v_j = squash(s_j) = \frac{\|s_j\|^2}{1+\|s_j\|^2} \frac{s_j}{\|s_j\|} \quad (8)$$

Encoders and decoders are the two main building blocks of the fully trainable CapsNet architecture. The encoder consists of three layers:

- A convolutional tier with ReLU activation (ReLU Conv1).
- A PrimaryCaps tier combining convolutional and capsule operations.
- A DigitCaps level, which generates vector representations for each output category.

The decoder receives a 16-dimensional vector output via DigitCaps level corresponding to the actual class tag. All other capsule outputs are masked with zeros. In contrast to CNNs, CapsNet exhibits translational equivariance rather than invariance, enabling better generalization across numerous perspectives using fewer training data. Moreover, the CapsNet

and LSTM models can be integrated in a manner similar to the CNN-LSTM architecture, where CapsNet sequentially extracts high-level capsule features from a series of data, and the LSTM layer generates an estimation of the target variable based on these extracted features.

3.2.2. Self-Attention (SA)

In this sub-section, an SA mechanism is incorporated to the CLSTM model instead of using a traditional convolution layer (input to primary capsule layer) with SA convolution layer. The reason for this is to apply boosting to the importance of features within the system at crucial stages to relatively boost the efficiency of the system in capturing contextual relationships through the filtering of noise factors. The idea here is to prioritize the importance of different regional styles of training in the input flow and to update these weights iteratively using the key and value characteristic matrices. Two different types of pooling transformations Max Pooling (MaxPool) and Average Pooling (AvgPool) were used to integrate the stream data of the features map.

The SA process is formulated as follows:

$$c = f_{conv} \cdot f_{Attention} = f_{conv}(softmax(K', Q') \cdot V) = f_{conv}(softmax(K.(f_{conv}[MaxPool(K); AvgPool(K)]) \cdot Q \cdot (f_{conv}[MaxPool(Q); AvgPool(Q)]))) \quad (9)$$

Where K, Q, V represent Key, Query and Value matrix.

The Primary Capsule layer receives the feature maps produced by the SA convolution layer as input. This layer converts scalar neurons into vectorized capsules of dimension 8, preserving both presence and pose information of features. Each capsule encodes spatial and contextual information from the input sequence, enabling the network to capture local relationships between features more effectively than traditional CNN layers. The capsules in this tier are linked to the Digit Capsule level via routing changes, which iteratively update the coupling coefficients among lowest and the highest-level capsules.

The routing process ensures that each low-level capsule serves appropriately to the high-level capsules based on agreement scores. The digital capsule phase conveys attributes to the LSTM tier, which is subsequently delivered to the entire linked layer, which aggregates the information to form higher-level representations for classification or prediction tasks. Here, ReLu is used as the activation function. The outcome of the completely linked layer may be obtained as

$$out(t) = [out_1^t, out_2^t, \dots, out_n^t] \quad (10)$$

Figure 2 shows the architecture of the CLSTM-SA model.

3.3. Client Model Generation-based Client Similarity in Server

The previous Sections (3.1 & 3.2) described the primary techniques used to extract and process raw data from the clients. Once the PPCFGAN models are trained for n clients, they are shared with the server for aggregation, enabling collaborative model updates. On the server side, each shared model generates a synthetic IID dataset, which helps to reduce the impact of non-IID data. Following these, a pre-trained AE is then employed to extract latent embeddings for each client, which are used to measure similarity and perform clustering in the cloud layer.

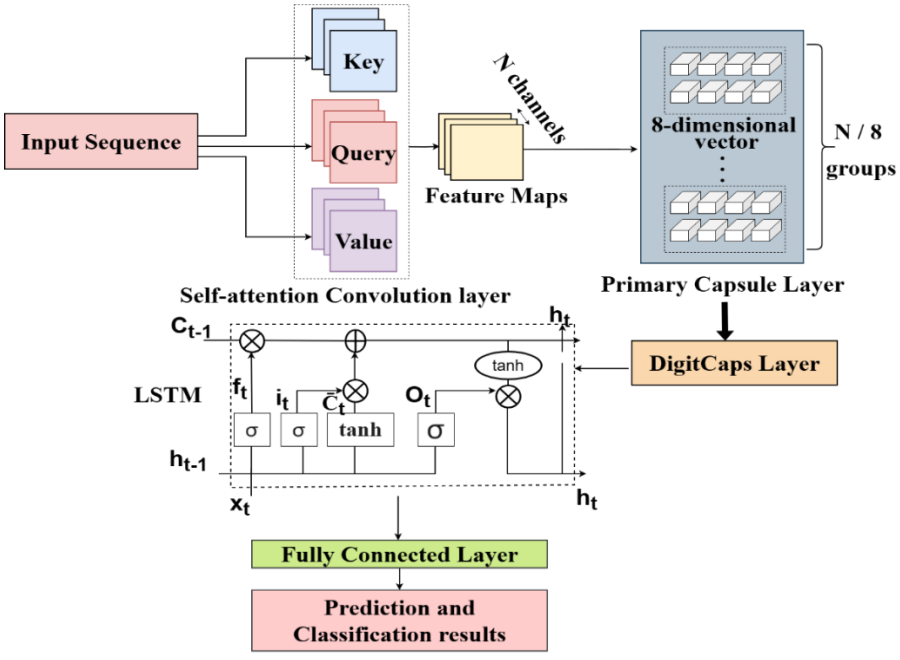

Fig. 2 Architecture of the CLSTM-SA model

3.3.1. Autoencoder for Client Similarity

In this sub-section, a pre-trained AE with an Encoder (E) and a Decoder (D) is employed to measure similarity between clients. Let X_i denote the private data distribution of client i, where $i = 1,2, \dots, n$. The encoder maps each training sample to a low-dimensional latent representation l:

$$\mathcal{E}(x_j) \mapsto l_j \tag{11}$$

The latent representation l_j is converted to a probability vector using the Softmax function:

$$p_j = Softmax\left(l_j\right) \tag{12}$$

Here, p_j represents the probability distribution of each feature in the encoder's latent space for the corresponding sample. For client i, the probability distribution over all samples is computed as:

$$P_i = \frac{\sum_{x \in X_i} \mathcal{E}(x)}{\sum_{k=1}^{R} \sum_{x \in X_i} \mathcal{E}(x)} \tag{13}$$

Finally, the distance between clients i and j is measured using a similarity matrix. Here, Kullback-Leibler Divergence (KLD) is employed for calculating similarity, which is computed as:

$$d_{ij} = KLD(P_i \| P_j) \tag{14}$$

These similarity distances are then utilized to cluster clients with comparable data distributions within the cloud server layer. Following this, a weighted collaborative aggregation strategy is applied within each cluster. In contrast to the existing approaches, where the data aggregation is weighted as all clients or weighted by the data size, in the PPCFGAN approach, clients cooperatively work in a cluster to create the aggregated result at the server, and they are weighted by client similarity. Clients who are more similar have larger weights in the cluster model, and less similar clients contribute with lower weights.

Aggregation generates a new dataset for each cluster of the PPCFGAN model. The server combines the synthetic data of different clients of the cluster to directly produce new data for the cluster. This data set distributes the data across multiple clients, rather than just one.

Finally, a cluster-specific PPCFGAN model is trained, which is subsequently shared with all clients in the corresponding cluster. Each client uses the received PPCFGAN model to generate additional synthetic samples, which are combined with their private local data to train downstream models locally, ensuring both personalization and privacy-preserving FL.

4. Results and Discussion

This part evaluates the efficiency of proposed PPCFGAN-CPDFedL model.

4.1. Dataset Description

Since the study focuses on addressing data heterogeneity, the evaluation procedure is crucial to demonstrate its effectiveness. To this end, two datasets from different domains are employed: An E-Commerce Data Warehouse dataset and a Healthcare Dataset: Heart Attack Analysis and Prediction. E-Commerce dataset [22, 23]: The dataset used in this project is the E-Commerce Data, which contains transactions from an online retail platform. It includes information like the invoice number, stock codes, definition, amount, invoice date, unit cost, consumer ID, and nationality. CIFAR-100 dataset [24]: The CIFAR-100 dataset comprises 60,000 color images, each of measuring 32 × 32 pixels, categorized into 100 distinct categories, with 600 images to each group. The dataset is categorized into 20 superclasses, where every image is annotated with dual tags: a "fine" tag representing the unique form, and a "coarse" tag representing the superclass to which it belongs. 50,000 out of these images are used for training, and 10,000 for testing.

4.2. Training Setup and Parameter Settings

The experimental analysis was carried out in the system with an operating system of Windows 10 64-bit running on Intel® CoreTM i5-4210 CPU (3 GHz), 4 GB of RAM and 1 TB hard disk. The proposed model's parameters are shown in Table 1.

4.3. Performance Metrics

For Generation:

1. Maximum Mean Discrepancy (MMD): It measures the difference between two probability measures, enabling comparison of their expected values in a Reproducing Kernel Hilbert Space (RKHS).

$$MMD^2(P, Q) = \left\|\mu_P - \mu_Q\right\|_{\mathcal{F}}^2 \tag{15}$$

2. Fréchet Inception Distance (FID) Score: It measures how close the produced images to authentic images based on feature statistics extracted from a pre-trained Deep Neural Network (DNN).

Table 1. Hyperparameter settings

Parameters	Range
Dropout Rate	0.3
Momentum	0.7
Epoch	100
Activation Function	ReLU
Weight Decay	0.0001
Learning Rate	0.0005
Optimizer	Adam

Batch size	32
Local Epochs (per client)	5
Number of Clients	20
DP Noise Multiplier	0.5
DP Parameters	ε = 1.0, δ = 1e-5

$$d^2((\mu_P, \Sigma_P), (\mu_Q, \Sigma_Q)) = \|\mu_P - \mu_Q\|_2^2 + Tr(\Sigma_P + \Sigma_Q - 2(\Sigma_P\Sigma_Q)^{1/2}) \quad (16)$$

For Classification: To demonstrate the efficiency of the proposed system (model), the following four traditional classification criteria are used. They are:

1. Accuracy: It is the ratio of accurately predicted instances to the total number of instances. It is defined as:

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \quad (17)$$

Where TP, TN, FP, and FN represent True Positives, True Negatives, False Positives, and False Negatives, respectively.

2. Precision: It measures the framework's ability to identify positive cases among all expected positive instances. The higher the precision, the fewer the false positives.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (18)$$

3. Recall or Sensitivity: It quantifies the ratio of true positives accurately recognized by the framework. It demonstrates the model's capacity to identify pertinent scenarios.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (19)$$

4. F-measure: The F-measure represents the harmonic mean of Precision and Recall, offering a fair assessment of both measurements. It is especially beneficial when the dataset is imbalanced.

$$\text{F} - \text{measure} = 2 \times \frac{Precsion \times Recall}{Precision + Recall} \quad (20)$$

4.4. Performance Evaluation

This section evaluates the proposed model in various ways.

4.4.1. Evaluation of Proposed PPCFGAN with Standard GAN Models

The proposed PPCFGAN is evaluated against various baseline GAN models like Federated GAN (FedGAN), Federated WGAN (FedWGAN), Federated WGAN-GP (FedWGAN-GP), Federated Conditional GAN (FedCGAN). To evaluate these models, two types of datasets are used, which is already described in Section 4.1. The first dataset contains numerical data, whereas the second dataset contains image data. The reason for such selection is to demonstrate the ability of the model to address data heterogeneity across different data modalities. The models are evaluated using MMD, as described in Section 4.3. It measures the similarity among actual and produced data distributions. A lesser MMD value indicates that the produced instances are statistically nearer to the actual data, demonstrating better generative performance. Since the first dataset is in tabular form, each sample represents a numerical feature set. Before applying MMD, all these features are normalized to [0,1] to ensure uniformity.

The kernel-based distance is then computed directly on the numerical features. Figure 3 shows the comparison of federated GAN models on the E-commerce dataset. Next, the other dataset is also evaluated against the same using the same metric, as MMD scores also can be computed directly from the images. Figure 4 shows the MMD comparison of federated GAN models in CIFAR-100 dataset. Also, compare federated GAN models using another metric called the FID score (only for CIFAR-100, as FID is calculated only for image data). Figure 5 shows the FID comparison of federated GAN models on E- commerce dataset. Figure 6 shows the FID comparison of federated GAN models on CIFAR-100 dataset.

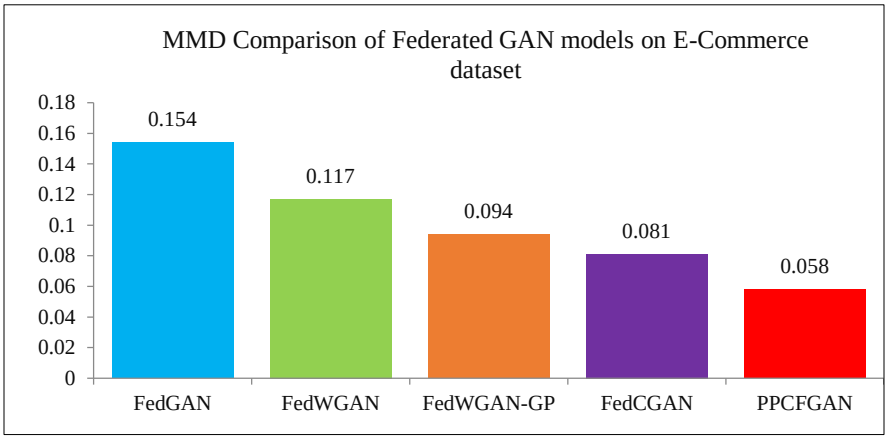

Fig. 3 MMD comparison of proposed model with existing federated GAN models on e-commerce dataset

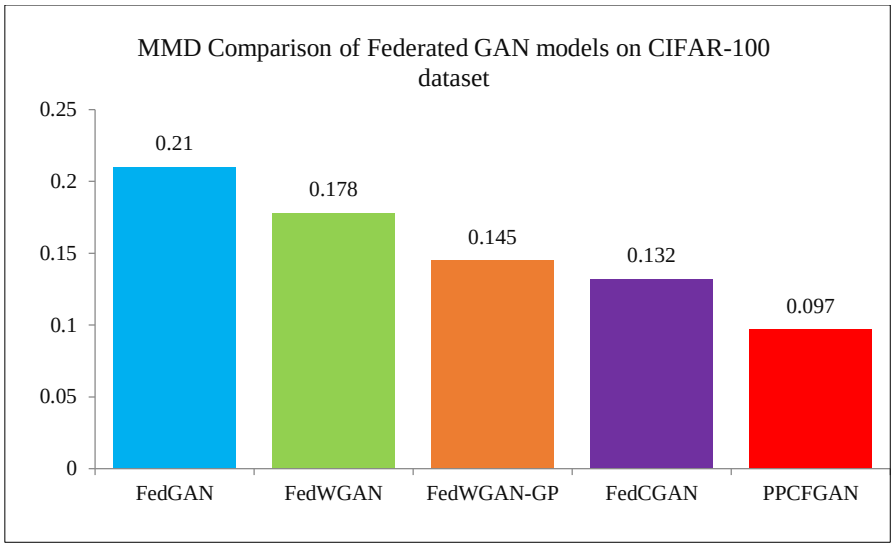

Fig. 4 MMD comparison of proposed model with existing federated GAN models on CIFAR-100 dataset

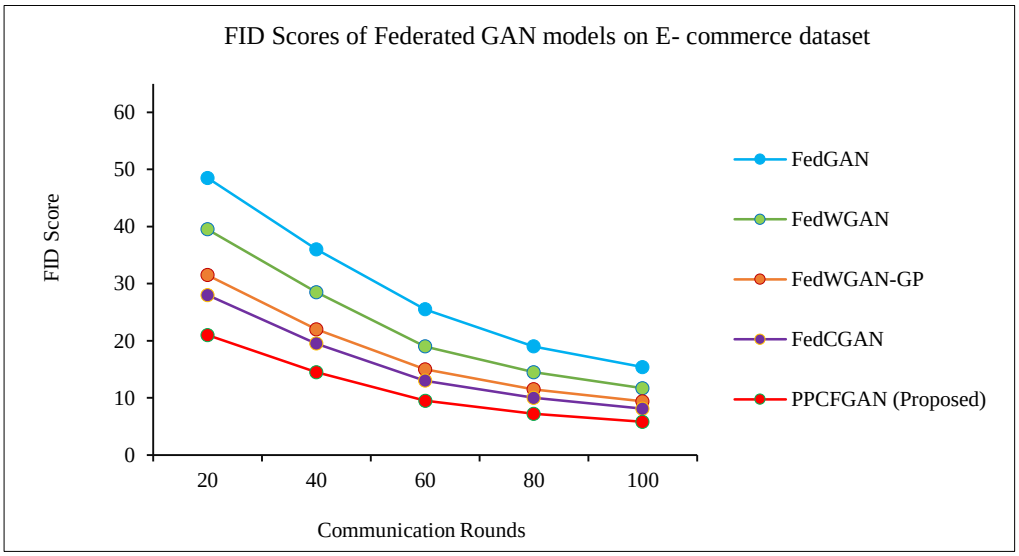

Fig. 5 FID comparison of proposed model with existing federated GAN models on e-commerce dataset

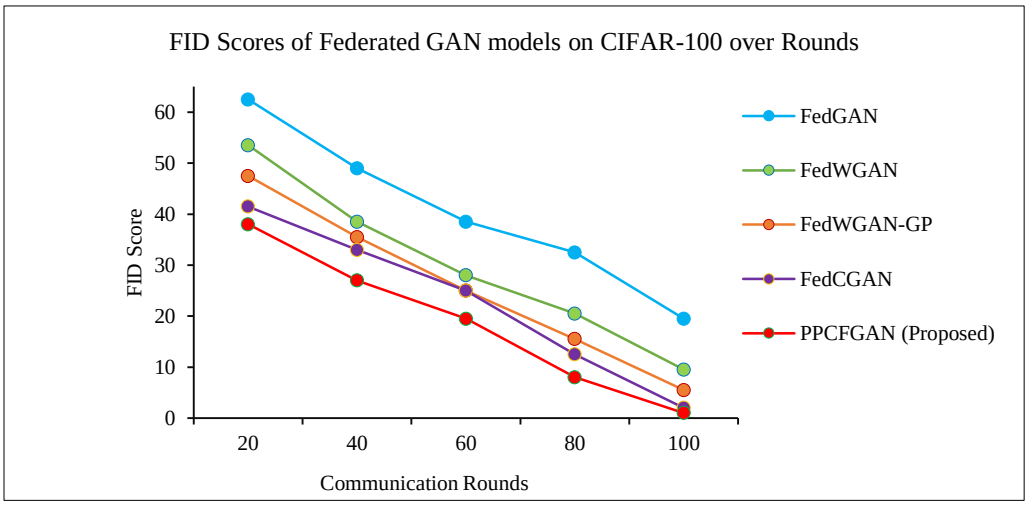

Fig. 6 FID comparison of proposed model with existing federated GAN models on CIFAR-100 dataset

4.4.2. Evaluation of Proposed PPCFGAN-CPDFedL Model-Group Non-IID and Client Non-IID Scenario

The proposed PPCFGAN-CPDFedL model is evaluated to analyze its performance under different configurations, specifically by varying the number of clusters and the number of clients.

Experiments are conducted on both E-commerce and CIFAR-100 datasets to validate the effectiveness of the model across different data types. The results demonstrate how these factors influence the learning capability and generalization performance of the proposed model.

Impact of Different Quantities of Clusters:

Figure 7 and Figure 8 show that a systematic scale-up in the number of clusters (k) improves final test performance and convergence rate with Group Non-IID and Client Non-IID distributions. In Figure 7, Clusters in the e-commerce dataset increase the final accuracy of 75% t to 93.26% by expanding clusters from k=2 and k=10, completely outpacing Standard FedAvg (k=1), which plateaus at a fluctuating 60% due to conflicting global gradients under this Group Non-IID and Client Non-IID setting.

In Figure 8, on the CIFAR-100 dataset, expanding the cluster from k=2 to k=10, accuracy increases from 62% to 94.13%. Finally, dividing heterogeneous clients into groups makes all harmful gradient conflicts irrelevant, turning Group Non-IID and Client Non-IID data distributions to localized homogeneity and significantly speeding up optimization performance compared to conventional global model aggregation.

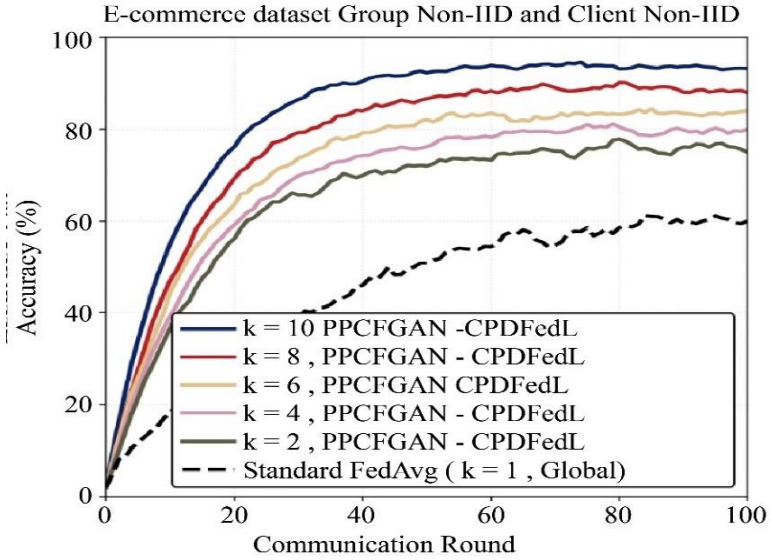

Fig. 7 The impact of different numbers of clusters with fixed total clients 100 on e-commerce dataset

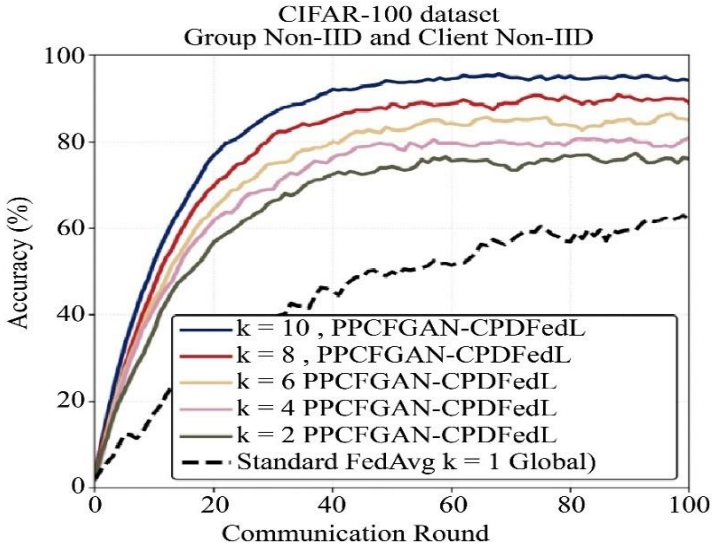

Fig. 8 The impact of different numbers of clusters with fixed total clients 100 on CIFAR-100 dataset

Different Numbers of Clients

Figure 9 and Figure 10 show that a systematic scale-up of total number of participating clients (N) with a fixed cluster-size (k=10) would systematically increase the final test accuracy and convergence under Group Non-IID and Client Non-IID distributions. In Figure 9, expansion of client population from N=20 to N=100 on the e-commerce dataset increases the final accuracy from 76% to 93.26%, that illustrates greater data diversity and cooperative representation of the clustered models in Group Non-IID and Client Non-IID case. In Figure 10, on the CIFAR-100 dataset, the N=100 setup provides the best and smoother performance curve, reaching an accuracy of 94.13%, smaller scales of clients, such as N=20, reach an accuracy of 77% and exhibit perhaps more noticeable local variations, given a fixed heterogeneity. Finally, the scale of clients in a fixed set of localized groups guarantees exploration of a wider range of features, reducing the effect of highly adverse Group Non-IID and Client Non-IID data distributions and reducing significantly the global optimization stability in both datasets.

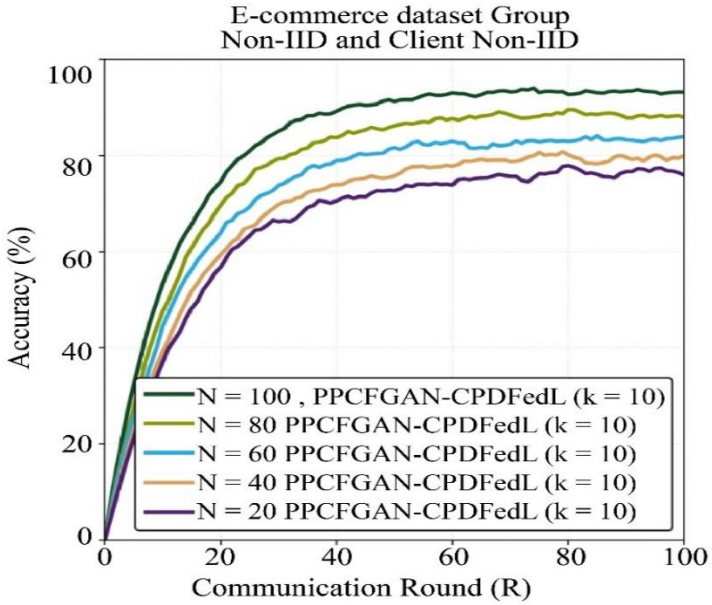

Fig. 9 The impact of different number clients with fixed cluster size 10 on e-commerce dataset

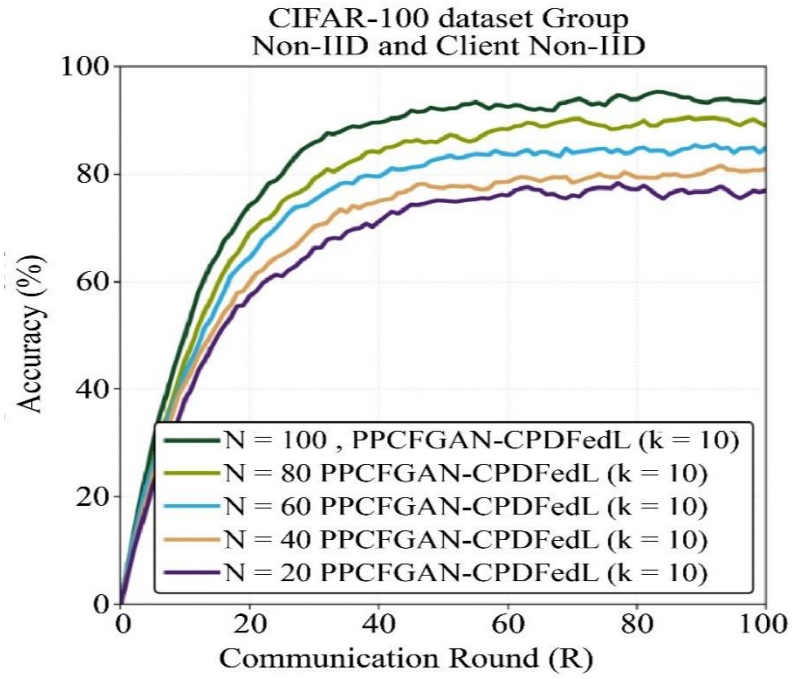

Fig. 10 The impact of different number clients with fixed cluster size 10 on CIFAR-100 dataset

4.4.3. Comparison of Proposed PPCFGAN-CPDFedL with Existing CPFL Models-Group Non-IID and Client Non-IID Scenario

This section evaluates the effectiveness of PCFGAN-CPDFedL model with existing models PGFL, PDP-FL, DP-CFL, pFedCAM for 10 cluster from 100 clients. Figures 11 and 12 demonstrate the performance of the various CPFL models and the proposed model in terms of accuracy in Group Non-IID and Client Non-IID conditions. The proposed PPCFGAN-CPDFedL model shows significant improvements over other models.

On the E-commerce dataset in Figure 11, the proposed model improves accuracy by 7.63% over PGFL, 6.17% over PDP-FL, 4% over DP-FL and 2.21% over pFedCAM. These findings show that PPCFGAN-CPDFedL significantly outperforms all other models, and its success in boosting model performance and generalization in different data distributions is observed. Likewise, under the CIFAR-100 dataset in Figure 12, the proposed model improves accuracy by 9.92% over PGFL, 7.59% over DP-FL, 4.57% over DP-CFL, and 2.59% over pFedCAM, respectively. This shows that the proposed PPCFGAN-CPDFedL model has a better performance than all the models currently available, showcasing its ability of enhancing model accuracy.

The comparative result of various CPFL models on the e-commerce dataset in terms of accuracy, precision, recall and F-measure is shown in Figure 13. The proposed PPCFGAN-CPDFedL model outperforming the other models in terms of all evaluation indicators. For instance, it increases accuracy by 7.63%, precision by 7.8%, recall by 8.78% and F-measure by 8.29% when compared to the vanilla PGFL. Compared to the PDP-FL, it increases accuracy by 6.17%, precision by 5.79%, recall by 6.08% and F-measure by 5.93%. For DP-CFL, the proposed model increases 4% (accuracy), 3.21% (precision), 3.60% (recall) and 3.40% (F-measure). The proposed method still results in a gain of around 2.21% (accuracy), 2.18% (precision), 1.82% (recall) and 2% (F-measure) for pFedCAM. Likewise, the other models also perform better than the other models, indicating that the incorporation of PPCFGAN-CPDFL greatly improves the model's regularization and generalization.

The comparative result of various CPFL models on the CIFAR-100 dataset in terms of accuracy, precision, recall and F-measure is shown in Figure 14. In particular, it improves accuracy by 9.92%, precision by 8.77%, recall by 7.57%, and F-measure by 8.17% compared to the vanilla PGFL model. Compared to the PDP-FL, it increases accuracy by 7.59%, precision by 7.42%, recall by 6.28% and F-measure by 6.85%.

For DP-CFL, the proposed model increases 4.57% (accuracy), 6.26% (precision), 3.87% (recall) and 5.07% (F-measure). For pFedCAM, the proposed model increases 2.59% (accuracy), 3.84% (precision), 2.30% (recall) and 3.07% (F-measure). Similarly, the proposed model significantly outperformed every other model, demonstrating that the integration of PPCFGAN with CPDFL substantially enhances model generalization and stability.

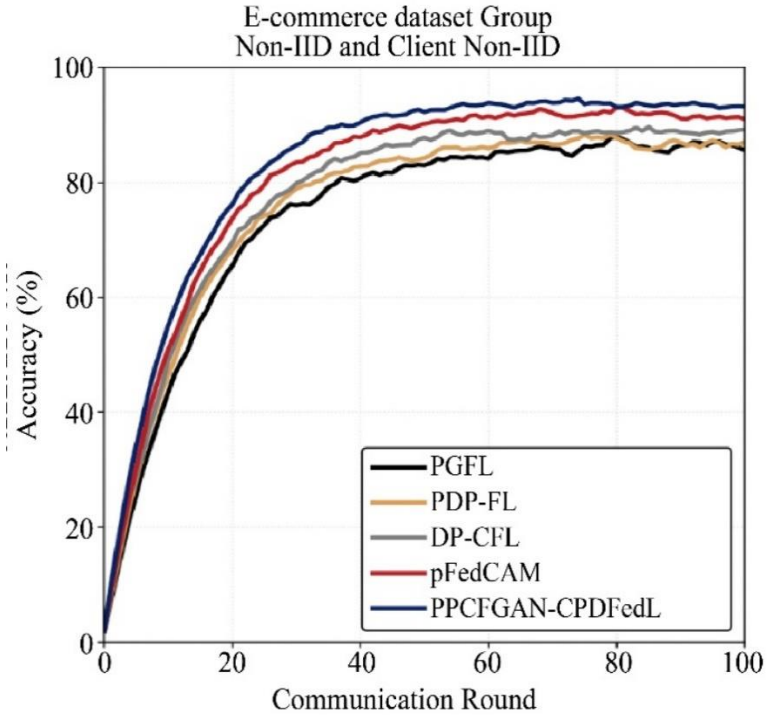

Fig. 11 Comparison of CPFL models on e-commerce dataset for group Non-IID and client Non-IID scenarios

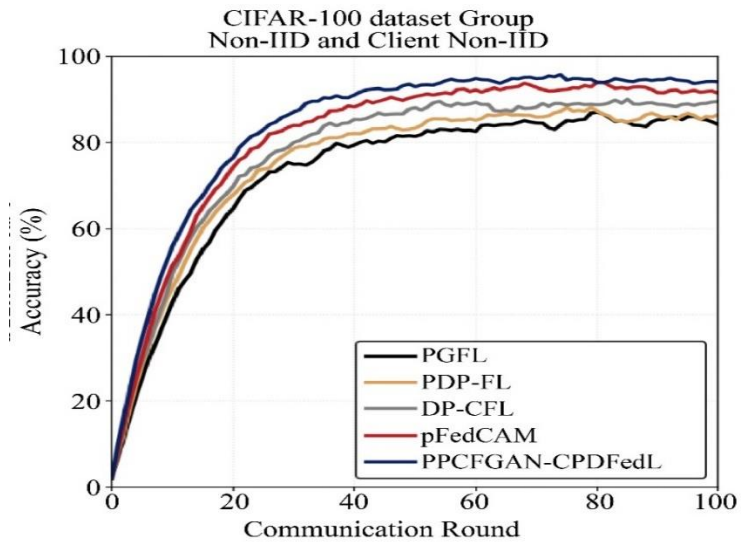

Fig. 12 Comparison of CPFL models on CIFAR-100 dataset for group Non-IID and client Non-IID scenarios

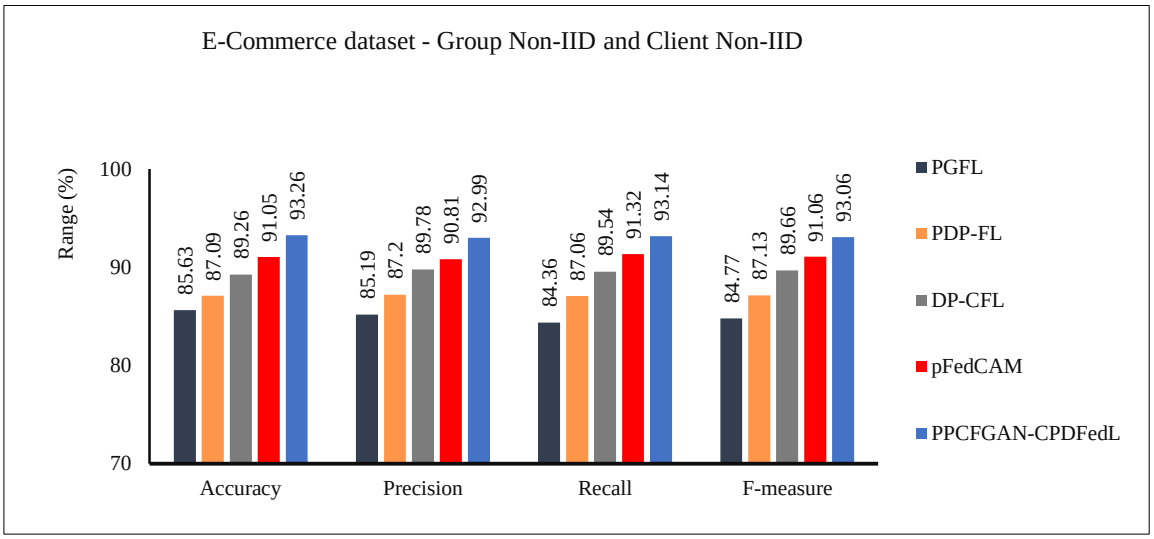

Fig. 13 Comparison of various metrics on e-commerce dataset for group Non-IID and client Non-IID scenario

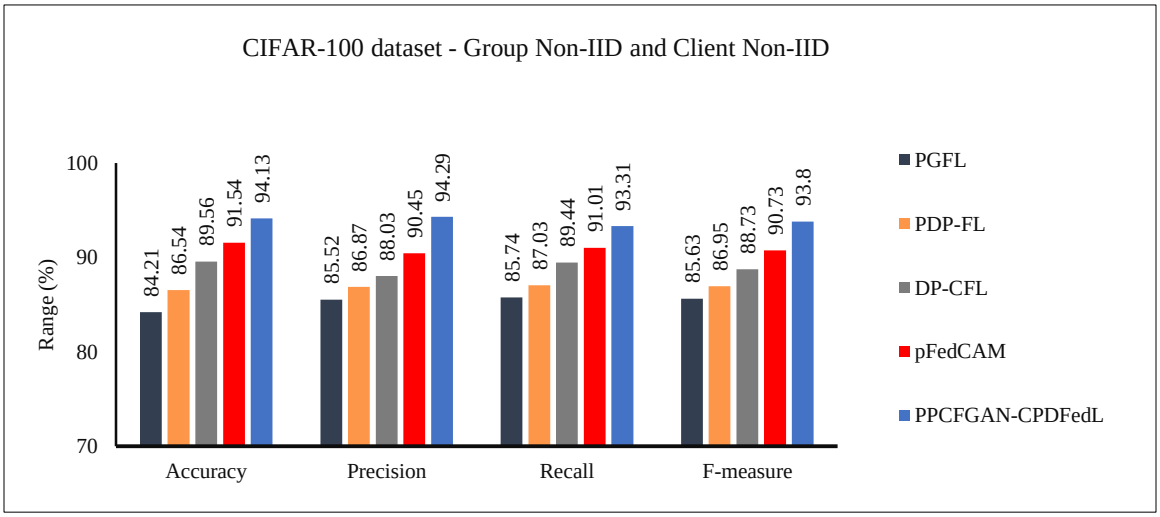

Fig. 14 Comparison of various metrics on CIFAR-100 dataset for group Non-IID and client Non-IID scenario

4.4.4. Comparison of Proposed PPCFGAN-CPDFedL with Existing CPFL Models-Group IID and Client IID Scenario

This section evaluates the effectiveness of PCFGAN-CPDFedL model with existing models PGFL, PDP-FL, DP-CFL, pFedCAM for 10 cluster from 100 clients. Figures 15 and 16 demonstrate the performance of the various CPFL models and the proposed model in terms of accuracy in Group IID and Client IID conditions. The proposed PPCFGAN-CPDFedL model shows significant improvements over other models. In Figure 15, on the e-commerce dataset, the proposed model improves accuracy by 9.25% over PGFL, 7.42% over PDP-FL, 5.15% over DP-CFL and 2.68% over pFedCAM.

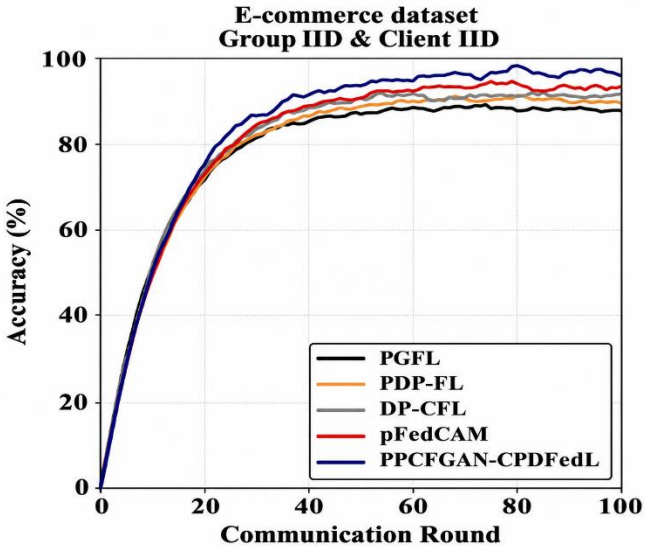

Fig. 15 Comparison of CFL models on e-commerce dataset for Group IID and Client IID scenarios

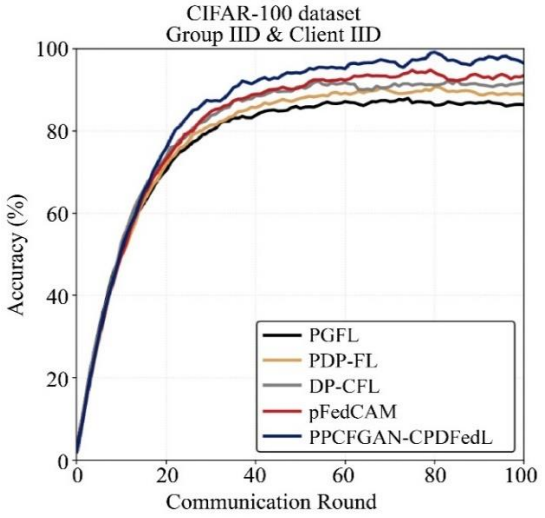

Fig. 16 Comparison of CFL models on CIFAR-100 dataset for Group IID and Client IID scenarios

These findings show that PPCFGAN-CPDFedL significantly outperforms all other models, and its success in boosting model performance and generalization in different data distributions is observed.

Figure 16, on the CIFAR-100 dataset, the proposed model improves accuracy by 10.45% over PGFL, 7.65% over DP-FL, 5.35% over DP-CFL, and 2.45% over pFedCAM, respectively.

This shows that the proposed model has a better performance than all the models currently available, showcasing its ability of enhancing model accuracy.

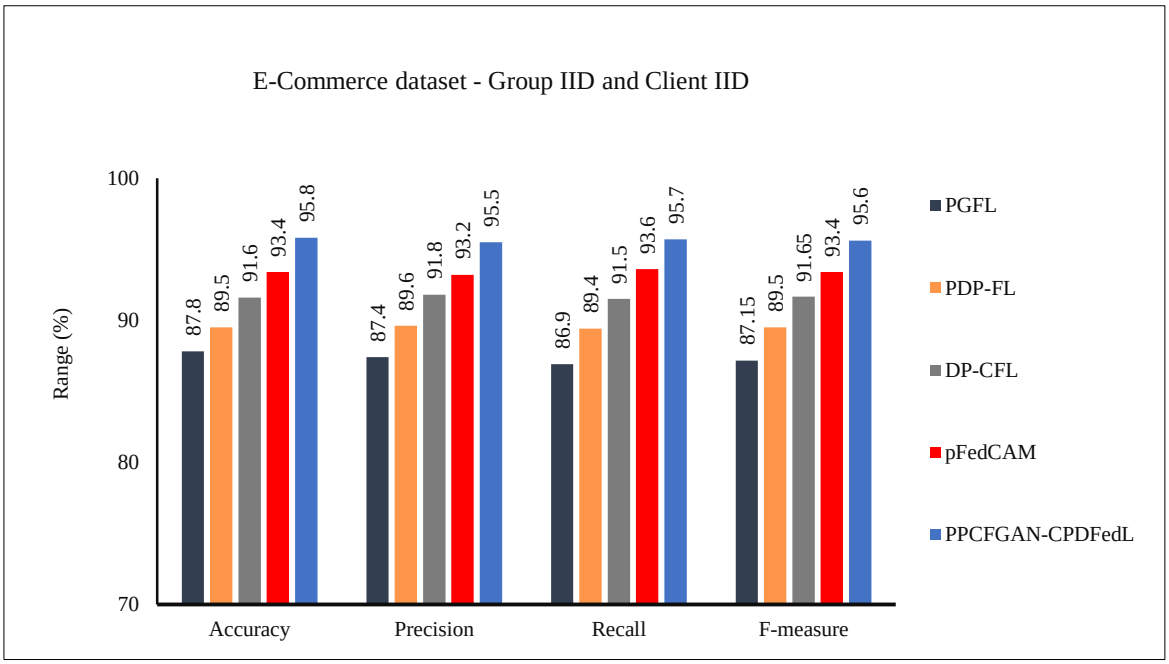

Fig. 17 Comparison of various metrics on e-commerce dataset for Group IID and Client IID scenario

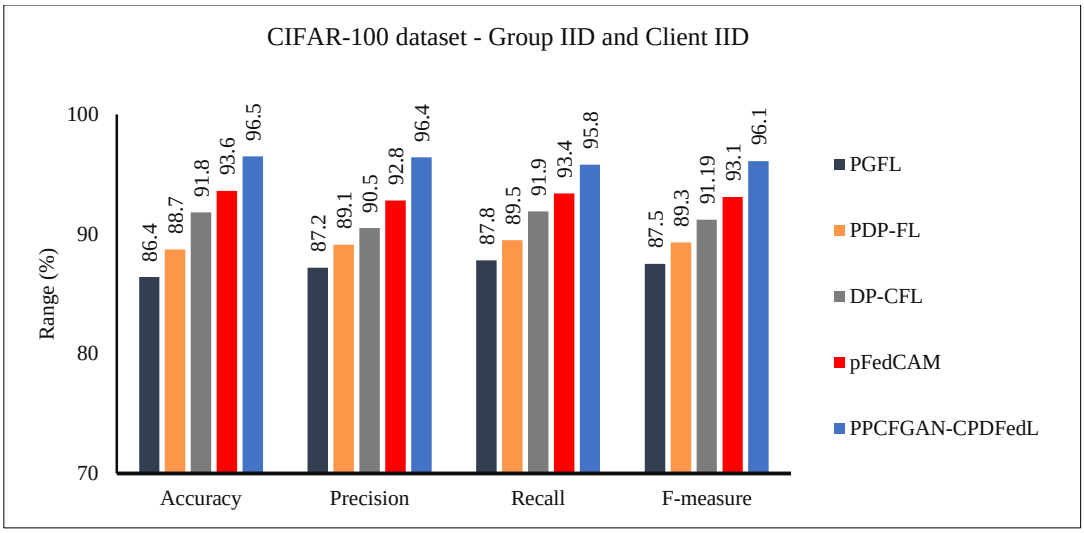

Fig. 18 Comparison of various metrics on CIFAR-100 dataset for Group IID and Client IID scenario

Figure 17 depicts the comparative performance of different CPFL frameworks on the e-commerce dataset under Group IID and Client IID environment. The suggested PPCFGAN-CPDFedL model ranks above all the baseline models in all evaluation indices. Namely, the proposed method improves the accuracy by 8%, precision by 8.10%, recall by 8.80%, and F-measure by 8.45%, compared to vanilla PGFL. For PDP-FL model, it increases accuracy by 6.30%, precision by 5.90%, recall by 6.30% and F-measure by 6.10%. Compared with DP-CFL, the proposed model achieves 4.20% in accuracy, 3.70% in precision, 4.20% in recall, and 3.95% in F-measure. Moreover, it has a strong performance advantage against the competitive pFedCAM framework, with an apparent enhancement of 2.40% in accuracy, 2.30% in precision, 2.10% in recall, and 2.20% in F-measure, hence affirming that the application of PPCFGAN can substantially strengthen model regularization. Figure 18 provides the performance analysis of the CPFL models under discussion with the help of fine- grained CIFAR-100 dataset under the Group IID and Client IID configuration. The proposed PPCFGAN-CPDFedL model is demonstrated to enhance its performance significantly over the basic baselines in all metric indicators. Specifically, it also increases accuracy (10.10%), precision (9.20%), recall (8%), and F-measure (8.60%) as compared to the usual PGFL baseline. For PDP-FL model, it increases accuracy by 7.80%, precision by 7.30%, recall by 6.30% and F-measure by 6.80%. Compared with DP-CFL, the proposed model achieves 4.70% in accuracy, 5.90% in precision, 3.90% in recall, and 4.91% in F-measure. In the case of the advanced pFedCAM model, the specified approach continues to provide concrete performance improvements of about 2.90% of accuracy, 3.60% of precision, 2.40% of recall and 3% of F-measure, which proves that the model can contribute significantly to the overall model generalization and stability.

5. Conclusion

This paper introduced a privacy-and-personalized PPCFGAN-CPDFedL model for diverse Cloud-IoT systems. The method employs a combination of PPCFGAN, WGAN-GP and DP at the client to generate privacy-preserved and valid synthetic data, while a CLSTM-SA classifier is used to boost the prediction accuracy. At the cloud, a pre-trained autoencoder supports similarity and weighted clustering of clients at the cloud and collaborative learning, to overcome issues of non-IID data distribution, heterogeneity and scalability. Our experimentation on E-Commerce and CIFAR-100 datasets shows PPCFGAN-CPDFedL consistently outperforms other approaches, reaching 93.26% and 94.13% accuracy, respectively and boosting precision, recall and F-measure for Group Non-IID and Client Non-IID scenario.

Conflicts of Interest

The author(s) declare(s) that there is no conflict of interest regarding the publication of this paper.

References

[1] Georgios Lampropoulos, Kerstin Siakas, and Theofylaktos Anastasiadis, "Internet of Things in the Context of Industry 4.0: An Overview," *International Journal of Entrepreneurial Knowledge*, vol. 7, no. 1, pp. 4-19, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[2] Daniel Poul Mtowe, Lika Long, and Dong Min Kim, "Low-Latency Edge-Enabled Digital Twin System for Multi-Robot Collision Avoidance and Remote Control," *Sensors*, vol. 25, no. 15, pp. 1-25, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[3] Awatef Issaoui, Jenny Örtensjö, and M. Sirajul Islam, "Exploring the General Data Protection Regulation (GDPR) Compliance in Cloud Services: Insights from Swedish Public Organizations on Privacy Compliance," *Future Business Journal*, vol. 9, no. 1, pp. 1-13, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[4] Betul Yurdem et al., "Federated Learning: Overview, Strategies, Applications, Tools and Future Directions," *Heliyon*, vol. 10, no. 19, pp. 1-24, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[5] Chunyan Li et al., "A Review of IoT Applications in Healthcare," *Neurocomputing*, vol. 565, pp. 1-12, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[6] Cristina Stolojescu-Crisan, Calin Crisan, and Bogdan-Petru Butunoi, "An IoT-based Smart Home Automation System," *Sensors*, vol. 21, no. 11, pp. 1-23, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[7] Hai Chen et al., "FG-RS: Capture user Fine-Grained Preferences through Attribute Information for Recommender Systems," *Neurocomputing*, vol. 458, pp. 195-203, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[8] Changchun Long et al., "Edge Computing Framework for Cooperative Video Processing in Multimedia IoT Systems," *IEEE Transactions on Multimedia*, vol. 20, no. 5, pp. 1126-1139, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[9] Karthik Shivashankar, Ghadi S. Al Hajj, and Antonio Martini, "Scalability and Maintainability Challenges and Solutions in Machine Learning: Systematic Literature Review," *arXiv preprint arXiv*, pp. 1-45, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[10] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek, "Clustered Federated Learning: Model-Agnostic Distributed Multitask Optimization Under Privacy Constraints," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 8, pp. 3710-3722, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[11] Avishek Ghosh et al., "An Efficient Framework for Clustered Federated Learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1-12, 2020. [Google Scholar] [Publisher Link]

[12] Jianfei Zhang, and Yongqiang Shi, "A Personalized Federated Learning Method based on Clustering and Knowledge Distillation," *Electronics*, vol. 13, no. 5, pp. 1-15, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[13] Mahtab Talaei, and Iman Izadi, "Adaptive Differential Privacy in Federated Learning: A Priority-based Approach," *arXiv preprint arXiv*, pp. 1-7, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[14] Yang Zhang et al., "DP-FedCMRS: Privacy-Preserving Federated Learning Algorithm to Solve Heterogeneous Data," *IEEE Access*, vol. 13, pp. 41984-41993, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[15] Francois Gauthier et al., "Personalized Graph Federated Learning with Differential Privacy," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 9, pp. 736-749, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[16] Junxu Liu et al., "Personalized Differentially Private Federated Learning without Exposing Privacy Budgets," *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 4140-4144, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[17] Zhikai Yang et al., "Personalized Federated Learning with Model Interpolation among Client Clusters and its Application in Smart Home," *World Wide Web*, vol. 26, no. 4, pp. 2175-2200, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[18] Saber Malekmohammadi, Afaf Taik, and Golnoosh Farnadi, "Differentially Private Clustered Federated Learning," *arXiv preprint arXiv*, pp. 1-29, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[19] Yuntao Wang et al., "Social-Aware Clustered Federated Learning with Customized Privacy Preservation," *IEEE/ACM Transactions on Networking*, vol. 32, no. 5, pp. 3654-3668, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[20] Tingzhe Pan et al., "Differentially Private Clustered Federated Load Prediction based on the Louvain Algorithm," *Algorithms*, vol. 18, no. 1, pp. 1-18, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[21] Dongyu Wei et al., "Optimizing Communication and Device Clustering for Clustered Federated Learning with Differential Privacy," *IEEE Transactions on Mobile Computing*, vol. 25, no. 1, pp. 419-433, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[22] E-Commerce Data Warehouse, GitHub, 2026. [Online]. Available: https://github.com/mustachemo/ecommerce-data-warehouse

[23] Carrie, E-Commerce Dataset, Kaggle, 2026. [Online]. Available: https://www.kaggle.com/datasets/carrie1/ecommerce-data

[24] CIFAR-100 Dataset, 2026. [Online]. Available: https://cave.cs.toronto.edu/kriz/cifar.html