SCIENCE CHINA
Information Sciences

• **Supplementary File** •

DP-GenFL: A Local Differentially Private Federated Learning System through Generative Data

Jun Li[1], Kang Wei[1], Chuan Ma[2,1,3*] & Feng Shu[4]

[1]*School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing* 210000, *China;*
[2]*Zhejiang Lab, Hangzhou* 310000, *China;*
[3]*Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, Nanjing* 210000, *China;*
[4]*School of Information and Communication Engineering, Hainan University, Haikou* 570228, *China*

Appendix A Preliminary

Appendix A.1 Federated learning

Let us consider a general FL system consisting of one server and N clients. Let $\mathcal{D}_i$ denote the local database held by the client $\mathcal{C}_i$, where $i \in \{1, 2, \ldots, N\}$. At the server, the goal is to learn a model over data that resides at the N associated clients. An active client, participating in the local training, needs to find a vector $\boldsymbol{w}$ of an AI model to minimize a certain loss function. Formally, the server aggregates the weights received from the N clients as

$$\boldsymbol{w} = \sum_{i=1}^{N} p_i \boldsymbol{w}_i, \tag{A1}$$

where $\boldsymbol{w}_i$ is the parameter vector trained at the i-th client, $\boldsymbol{w}$ is the parameter vector after aggregating at the server, and N is the number of clients. In addition, we consider the standard aggregation rule that $p_i = \frac{|\mathcal{D}_i|}{|\mathcal{D}|} \geqslant 0$ with $\sum_{i=1}^{N} p_i = 1$, and $|\mathcal{D}| = \sum_{i=1}^{N} |\mathcal{D}_i|$ denotes the total size of all data samples. Such an optimization problem can be formulated as

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \sum_{i=1}^{N} p_i F_i(\boldsymbol{w}, \mathcal{D}_i), \tag{A2}$$

where $F_i(\cdot)$ is the local loss function of the i-th client. Generally, the local loss function $F_i(\cdot)$ is given by local empirical risks. The training process of such an FL system usually contains the following four steps:

• **Local training:** All active clients locally compute training gradients or parameters and send locally trained ML parameters to the server;

• **Model aggregating:** The server performs secure aggregation over the uploaded parameters from N clients without learning local information;

• **Parameters broadcasting:** The server broadcasts the aggregated parameters to the N clients;

• **Model updating:** All clients update their respective models with the aggregated parameters and test the performance of the updated models.

In the FL process, the N clients with the same data structure will collaboratively learn an ML model with the help of a cloud server. After a sufficient number of local training and update exchanges between the server and its associated clients, the solution to the optimization problem (A2) is able to converge to that of the global optimal learning model.

Appendix A.2 Differentially private FL

(ϵ, δ)-DP provides a strong criterion for privacy preservation of distributed data processing systems [1,2]. Here, $\epsilon > 0$ is the distinguishable bound of all outputs on neighboring datasets $\mathcal{D}_i, \mathcal{D}_i'$ in a database, and δ represents the event that the ratio of the probabilities for two adjacent datasets $\mathcal{D}_i, \mathcal{D}_i'$ cannot be bounded by e^{ϵ} after adding a privacy-preserving mechanism. With an arbitrarily given δ, a privacy-preserving mechanism with a larger ϵ gives a clearer distinguishability of neighboring datasets and hence a higher risk of privacy violation. Now, we will formally define DP as follows.

* Corresponding author (email: chuan.ma@njust.edu.cn)

Definition 1. ((ϵ, δ)-DP [1]): A randomized mechanism $\mathcal{M}: \mathcal{X} \to \mathcal{R}$ with domain $\mathcal{X}$ and range $\mathcal{R}$ satisfies (ϵ, δ)-DP, if for all measurable sets $\mathcal{S} \subseteq \mathcal{R}$ and for any two adjacent databases $\mathcal{D}_i, \mathcal{D}_i' \in \mathcal{X}$,

$$Pr[\mathcal{M}(\mathcal{D}_i) \in \mathcal{S}] \leqslant e^{\epsilon} Pr[\mathcal{M}(\mathcal{D}_i') \in \mathcal{S}] + \delta. \tag{A3}$$

For numerical data, a Gaussian mechanism defined in [1] can be used to guarantee (ϵ, δ)-DP. The conventional deep learning models attain DP guarantees via two alterations to the training process: the clipping of per-sample gradients, and the addition of Gaussian noise to gradients, as known as DP-SGD. The work [3] first applied DP in the FL system, denoted as DP-FL, which is the main counterpart of this paper. In the following algorithm, we show the main steps of the DP-FL procedure.

Algorithm A1 Local Differentially Private Federated Learning System through Additive noise (DP-FL)

Require: The initial global model $\boldsymbol{w}^{(0)}$, the maximum number of the communication round T, the random additive noise $N(0, \sigma^2)$;
Ensure: The trained global model $\boldsymbol{w}^f$;
Set the communication round index $t = 0$;
while $t \leqslant T$ **do**
  for Each client i in $\mathcal{C}$ **do**
    Train the local model $\boldsymbol{w}_i^{(t)}$;
    Clip the local gradient during the training process as $\boldsymbol{g}_i^{(t)} = \boldsymbol{g}_i^{(t)} / \max\left(1, \frac{||\boldsymbol{g}_i^{(t)}||}{C}\right)$;
    Add noise and upload models $\boldsymbol{g}_i^{(t)} = \boldsymbol{g}_i^{(t)} + \boldsymbol{n}_i^{(t)}$;
  end for
  Aggregate all received local models as: $\boldsymbol{w}^{(t+1)} = \frac{1}{N} \sum_{i \in \mathcal{N}} \boldsymbol{w}_i^{(t)}$;
  Increase the communication round index $t = t + 1$;
end while

Appendix A.3 Membership inference attack

Although the individual dataset $\mathcal{D}_i$ of the i-th client is kept locally in FL, the intermediate parameter $\boldsymbol{w}_i$ needs to be shared with the curious server, which may reveal the clients' private information as demonstrated by membership inference attacks. For example, authors in [4] proposed a shadow learning technique-based membership inference attack method, which can determine if a given data record was in the model's training dataset via black-box access to this model. As reported in [4,5], the proposed membership attack can achieve a nearly 73% and 58% accuracy on CIFAR-10 [6] and MNIST [1)] dataset, respectively. To prevent such an attack, [3,7] perturbed the trained model with additive noises to achieve DP and obtained a degradation in attacking rate with an irreparable sacrifice on learning performance. For example, a 15% (58%), and 5% (53%) reduction is obtained when $\epsilon = 4$.

Appendix B Algorithms

Appendix B.1 Algorithm B1: Conditional-GAN based data augmentation

The generator $\mathbb{G}$ and discriminator $\mathbb{D}$ is trained with the following loss function:

$$\min_{\mathbb{G}} \max_{\mathbb{D}} V\left(\mathbb{G}(\boldsymbol{z}, y), \mathbb{D}\left((\boldsymbol{x}, y), \mathbb{G}(\boldsymbol{z}, y)\right)\right), \tag{B1}$$

where $\boldsymbol{z}$ is a random vector, and $V(\cdot, \cdot)$ is a distance measurement between distributions parameterized by $\mathbb{G}$ and $\mathbb{D}$. In this paper, we use Jensen-Shannon divergence with the formula:

$$\begin{aligned} V\left(\mathbb{G}(\boldsymbol{z}, y), \mathbb{D}\left((\boldsymbol{x}, y), \mathbb{G}(\boldsymbol{z}, y)\right)\right) &= \mathbb{E}_{\boldsymbol{x} \sim p(\boldsymbol{x})}[\log \mathbb{D}(\boldsymbol{x}, y)] \\ &+ \mathbb{E}_{\boldsymbol{z} \sim p(\boldsymbol{z})}[\log(1 - \mathbb{D}(\mathbb{G}(\boldsymbol{z}, y)))]. \end{aligned} \tag{B2}$$

Via this loss, we can generate the required data samples in the conditioned labels.

The procedure of the proposed conditional-GAN-based data augmentation has been presented in **Algorithm B1**. At the beginning of this algorithm, each client will receive the required number of training datasets. First, each client trains the generator $\mathbb{G}$ based on the conditional-GAN based on its unique training dataset $\mathcal{D}_i$. Then, each client calculates the ratio for each class, i.e., q_m, $\forall m \in \mathcal{M}$. Afterward, each client generates $q_m * n$ data samples from the trained $\mathbb{G}$. We can notice that this algorithm aims to generate a dataset $\widetilde{\mathcal{D}}_i$ containing n samples as well as the same class ratio as the training dataset $\mathcal{D}_i$ for each client.

Appendix B.2 Algorithm B2: Local Differentially Private Federated Learning System through Generative Data

1) http://yann.lecun.com/exdb/mnist

Algorithm B1 Conditional-GAN based Data Augmentation

Require: The ith client's training dataset $\mathcal{D}_i$, $\forall i \in \mathcal{C}$, the number of generated data samples n and $\frac{n}{|\mathcal{D}_i|} \in \mathbb{Z}^+$;
Ensure: The ith client's generated dataset $\widetilde{\mathcal{D}}_i$, $\forall i \in \mathcal{C}$;
1: Train the generator g based on the conditional-GAN with the input dataset $\mathcal{D}_i$;
2: Calculate the ratio for each class, i.e., q_m, $\forall m \in \mathcal{M}$;
3: **for** m from 1 to M **do**
4: Generate $q_m * n$ data samples from the trained $\mathbb{G}$;
5: **end for**
6: Combine all classes of the generated data and obtain the dataset $\widetilde{\mathcal{D}}_i$;

Algorithm B2 Local Differentially Private Federated Learning System through Generative Data (DP-GenFL)

Require: The LDP requirement for the ith client (ϵ_i, δ_i), $\forall i \in \mathcal{C}$, the initial global model $\boldsymbol{w}^{(0)}$, the maximum number of the communication round T;
Ensure: The trained global model $\boldsymbol{w}^f$;
1: Set the communication round index $t = 0$;
2: **while** $t \leqslant T$ **do**
3: **for** Each client i in $\mathcal{C}$ **do**
4: Generate the synthetic data $\widetilde{\mathcal{D}}_i$ using **Algorithm B1**;
5: Combine $\widetilde{\mathcal{D}}_i$ and $\mathcal{D}_i$ with ratios γ and $1-\gamma$, respectively, to generate $\widehat{\mathcal{D}}_i$ as the local training dataset;
6: Set the additive noise STD σ_i for the DP-FL;
7: Train the local model $\boldsymbol{w}_i^{(t)}$ with the sampled synthetic data $\widehat{\mathcal{D}}_i$;
8: **if** The sample used is the real data **then**
9: Train this sample using the DP-FL (**Algorithm A1**) mechanism;
10: **else**
11: Train this sample using the GD mechanism;
12: **end if**
13: Calculate the accuracy of the validation dataset $\mathcal{V}_i^{(t)}$;
14: Upload the local model $\boldsymbol{w}_i^{(t)}$ and the validation accuracy $\mathcal{V}_i^{(t)}$ to the server;
15: **end for**
16: Aggregate all received local models as: $\boldsymbol{w}^{(t+1)} = \frac{1}{N}\sum_{i\in\mathcal{N}} \boldsymbol{w}_i^{(t)}$;
17: Calculate the global validation accuracy as: $\mathcal{V}^{(t+1)} = \frac{1}{N}\sum_{i\in\mathcal{N}} \mathcal{V}_i^{(t)}$;
18: **if** $\boldsymbol{w}^{(t+1)}$ satisfies the requirement (up to the DP budget) or $t = T-1$ **then**
19: Set $\boldsymbol{w}^f = \boldsymbol{w}^{(t+1)}$;
20: **else**
21: Increase the communication round index $t = t+1$;
22: **end if**
23: **end while**

Appendix C Differential privacy analysis

Research in differentially private ML models tracks a relaxed variant of DP, known as Rényi-DP (RDP) [8] that has already been widely adopted, such as Opacus in Facebook and TensorFlow privacy in Google. Hence, we will adopt the RDP technique in this paper, and then formally define RDP as follows.

Definition 2. ((α, ϵ)-RDP): Given a real number $\alpha \in (1, +\infty)$, and privacy parameter ϵ, a randomized mechanism $\mathcal{M}$ satisfies (α, ϵ)-RDP for any two adjacent datasets $\mathcal{D}$, $\mathcal{D}' \in \mathcal{X}$, we have

$$\mathcal{D}_\alpha \left[\mathcal{M}(\mathcal{D})||\mathcal{M}(\mathcal{D}')\right] := \frac{1}{\alpha - 1} \log \mathbb{E}\left[\left(\frac{\mathcal{M}(\mathcal{D})}{\mathcal{M}(\mathcal{D}')}\right)^\alpha\right] \leqslant \epsilon, \tag{C1}$$

where the expectation is taken over the output of $\mathcal{M}(\mathcal{D})$.

We can note that RDP is a generalization of (ϵ, δ)-DP that adopts Rényi divergence as a distance metric between two distributions. It can be shown that pure (ϵ, δ)-DP is equivalent to (∞, ϵ)-RDP, and, further, that if a model $\mathcal{M}$ satisfies (α, δ)-RDP, then $\mathcal{M}$ also satisfies $(\epsilon + \frac{\log \frac{1}{\delta}}{\alpha - 1}, \delta)$-DP for any $\delta \in (0, 1)$.

Next, we show the main proof. Based on Theorem 21 in [2, 9] and the proposed algorithm, we have

$$\epsilon = T\tau A_\alpha + \frac{\log\left(\frac{1}{\delta}\right) + \log(\alpha) + (\alpha - 1)\log\left(1 - \frac{1}{\alpha}\right)}{\alpha - 1}, \tag{C2}$$

where A_α denotes the Rényi distance between two distributions, given by

$$A_\alpha = D_\alpha((1 - q)\mu_0 + q(1 - \gamma)\mu_1, \mu_0) + q\gamma D_\alpha(\nu_1, \nu_0), \tag{C3}$$

where ν_0 and ν_1 are the sample distribution generated by $\mathcal{G}(x)$ and $\mathcal{G}(x')$, respectively, γ is the ratio of fake samples. Due to Jensen's inequality, we have if a function $f(x)$ is convex, $f(\mathbb{E}(x)) \leqslant \mathbb{E}(f(x))$. Hence, we can obtain

$$\begin{aligned} D_\alpha(\nu_1, \nu_0) &= \frac{1}{\alpha - 1} \log \int_{-\infty}^{+\infty} \nu_0(z) \left(\frac{\nu_0(z)}{\nu_1(z)}\right)^{\alpha - 1} dz \\ &\leqslant \frac{1}{\alpha - 1} \int_{-\infty}^{+\infty} \nu_0(z) \log \left(\frac{\nu_0(z)}{\nu_1(z)}\right)^{\alpha - 1} dz \\ &= \int_{-\infty}^{+\infty} \nu_0(z) \log \left(\frac{\nu_0(z)}{\nu_1(z)}\right) dz = D_{\mathrm{KL}}(\nu_0, \nu_1). \end{aligned} \tag{C4}$$

Following the lemma 4 in [10, 11], we have

$$\max\{D_{\mathrm{KL}}(\nu_0, \nu_1), D_{\mathrm{KL}}(\nu_1, \nu_0)\} \leqslant \Gamma_{\mathcal{F},\mathcal{G}} \left(\tau_{k,\xi} + \frac{2\Delta}{m}\right), \tag{C5}$$

where $\Gamma_{\mathcal{F},\mathcal{G}} \triangleq \sup_{\nu_1,\nu_2 \in \mathcal{G}} \|\log(\rho_{\nu_1}/\rho_{\nu_2})\|_{\mathcal{F},1}$, $\mathcal{G}$ is the possible generators, $\mathcal{F}$ is the possible discriminators, ρ_{ν_1} and ρ_{ν_2} denote the density functions, $\tau_{k,\xi} = 2(\inf_{\nu \in \mathcal{G}} d_{\mathcal{F}}(\mu, \nu) + \tau_{\mathrm{opt}} + C_\xi/\sqrt{k})$, τ_{opt} is the optimization error, $C_\xi = 16\sqrt{2\pi}pL + 2\Delta\sqrt{2\log(1/\xi)}$, k is the number of sampled training samples used in training, m is the number of training datasets in GAN training and Δ is the upper bound of $\|f(x)\|_\infty$, $f \in \mathcal{F}$. Therefore, we have

Theorem 1 (ϵ-LDP)**.** The privacy level of ϵ-LDP for each client can be expressed as

$$\epsilon \leqslant \underbrace{D_\alpha((1 - q)\mu_0 + q(1 - \beta)\mu_1, \mu_0)}_{\text{Due to DP-FL}} + \underbrace{q\gamma\Gamma_{\mathcal{F},\mathcal{G}} \left(\tau_{k,\xi} + \frac{2\Delta}{m}\right)}_{\text{Due to generative samples}}. \tag{C6}$$

Remark 1. From Eq. (C6), the privacy level consists of two parts, i.e., the random perturbation on the uploaded models, and the generative fake samples. In addition, we provide a general expression of $D_\alpha((1 - q)\mu_0 + q(1 - \gamma)\mu_1, \mu_0)$ here, the exact expression usually needs integral approximation, which can be found in Section 3.3 of [8].

Appendix D Proof of Theorem 2

Appendix D.1 Assumptions

In this section, we present our theoretical results on the guarantees of the DP-GenFL algorithm considering non-convex settings. Before that, we first mention three customary assumptions required for non-convex settings.

Assumption 1. We make assumptions on the global loss function $F(\cdot)$ defined by $F(\cdot) \triangleq \sum_{i=1}^N p_i F_i(\cdot)$, and the i-th local loss function $F_i(\cdot)$ as follows:

1) $F_i(\boldsymbol{w})$ is ρ-Lipschitz smooth, i.e., $\|\nabla F_i(\boldsymbol{w}) - \nabla F_i(\boldsymbol{w}')\| \leqslant \rho\|\boldsymbol{w} - \boldsymbol{w}'\|$, for any $\boldsymbol{w}$, $\boldsymbol{w}'$, where ρ is a constant determined by the practical loss function;
2) For any i and $\boldsymbol{w}$, $\mathbb{E}\{\|\nabla F_i(\boldsymbol{w}) - \nabla F(\boldsymbol{w})\|\} \leqslant \varepsilon_i$, where ε_i is the divergence metric.
3) For any i and $\boldsymbol{w}$, the stochastic gradients of ${}_i(\boldsymbol{w}, \mathcal{D}_i)$ and $F_i(\boldsymbol{w}, \mathcal{D}_i')$ have the bounded variances, respectively, i.e.,

$$\mathbb{E}\{\|\nabla F_i(\boldsymbol{w}, \mathcal{D}_i) - \nabla F_i(\boldsymbol{w})\|\} \leqslant \widetilde{\sigma}_R/\sqrt{|\mathcal{D}_i|}, \tag{D1}$$

$$\mathbb{E}\{\|\nabla F_i(\boldsymbol{w}, \mathcal{D}_i') - \nabla F_i(\boldsymbol{w})\|\} \leqslant \widetilde{\sigma}_S/\sqrt{|\mathcal{D}_i'|}, \tag{D2}$$

where $\widetilde{\sigma}_R \geqslant \widetilde{\sigma}_S > 0$ and when the loss of conditional-GAN is 0, we have $\widetilde{\sigma}_R = \widetilde{\sigma}_S$.

Appendix D.2 Bounded variance for the generated data

Lemma 1 (Bounded variance for generated data)**.** For the fake samples generated in Algorithm B1, we can bound the generated data as

$$\mathbb{E}\left\{\left\|\nabla F_i\left(\boldsymbol{w},\widehat{\mathcal{D}}_i\right)-\nabla F(\boldsymbol{w})\right\|\right\}\leqslant\varepsilon_i+\widetilde{\sigma}_R\sqrt{\frac{(1-\gamma)}{D_{in}}}+\widetilde{\sigma}_S\sqrt{\frac{\gamma}{D_{in}}}. \tag{D3}$$

Proof. Based on Algorithm B1, the generated data can be expressed as $\widehat{\mathcal{D}}_i=f_{\text{sample}}(\mathcal{D}_i,1-\gamma)\bigcup f_{\text{sample}}(\widetilde{\mathcal{D}}_i,\gamma)$, where f_{sample} is the sample function and γ is the sample ratio. Therefore, we have

$$\begin{aligned}
\mathbb{E}\left\{\left\|\nabla F_i\left(\boldsymbol{w},\widehat{\mathcal{D}}_i\right)-\nabla F(\boldsymbol{w})\right\|\right\}&=\mathbb{E}\left\{\left\|(1-\gamma)\nabla F_i\left(\boldsymbol{w},\mathcal{D}_i\right)+\gamma\nabla F_i\left(\boldsymbol{w},\widetilde{\mathcal{D}}_i\right)-\nabla F(\boldsymbol{w})\right\|\right\}\\
&=\mathbb{E}\left\{\left\|(1-\gamma)\nabla F_i\left(\boldsymbol{w},\mathcal{D}_i\right)-(1-\gamma)\nabla F(\boldsymbol{w})+\gamma\nabla F_i\left(\boldsymbol{w},\widetilde{\mathcal{D}}_i\right)-\gamma\nabla F(\boldsymbol{w})\right\|\right\}\\
&\overset{(a)}{\leqslant}(1-\gamma)\mathbb{E}\left\{\left\|\nabla F_i\left(\boldsymbol{w},\mathcal{D}_i\right)-\nabla F(\boldsymbol{w})\right\|\right\}+\gamma\mathbb{E}\left\{\left\|\nabla F_i\left(\boldsymbol{w},\widetilde{\mathcal{D}}_i\right)-\nabla F(\boldsymbol{w})\right\|\right\}\\
&\overset{(b)}{\leqslant}\varepsilon_i+(1-\gamma)\mathbb{E}\left\{\left\|\nabla F_i\left(\boldsymbol{w},\mathcal{D}_i\right)-\nabla F_i(\boldsymbol{w})\right\|\right\}+\gamma\mathbb{E}\left\{\left\|\nabla F_i\left(\boldsymbol{w},\widetilde{\mathcal{D}}_i\right)-\nabla F_i(\boldsymbol{w})\right\|\right\}\\
&\overset{(c)}{\leqslant}\varepsilon_i+\widetilde{\sigma}_R\sqrt{\frac{(1-\gamma)}{D_{in}}}+\widetilde{\sigma}_S\sqrt{\frac{\gamma}{D_{in}}},
\end{aligned} \tag{D4}$$

where D_{in} is the number of training dataset, i.e., $D_{in}=|\widehat{\mathcal{D}}_i|$, step (a) applies the triangle inequality, step (b) and (c) are due to the second and third assumptions in **Assumption 1**, respectively.

Appendix D.3 Convergence bound of the proposed algorithm

Theorem 2 (Convergence bound)**.** The convergence loss for the proposed DP-GenFL (Algorithm B2) can be can given as

$$\frac{1}{NT\tau}\sum_{t=0}^{T-1}\sum_{i=1}^{N}\sum_{\ell=0}^{\tau-1}\left\|\nabla F_i\left(\boldsymbol{w}_i^{(t,\ell)}\right)\right\|^2\leqslant\frac{2\left(F\left(\boldsymbol{w}^{(0)}\right)-F\left(\boldsymbol{w}^{\star}\right)\right)}{\eta(1-\eta^2\rho^2\tau^2)}+\frac{\tau(1+\rho\eta\tau)}{1-\eta^2\rho^2\tau^2}\sum_{i=1}^{N}\sum_{\ell=0}^{\tau-1}\left(\varepsilon_i+\widetilde{\sigma}_R\sqrt{\frac{(1-\gamma)}{D_{in}}}+\widetilde{\sigma}_S\sqrt{\frac{\gamma}{D_{in}}}\right)^2. \tag{D5}$$

Proof. First, using the ρ-Lipschitz smooth, we have

$$\mathbb{E}\left\{F\left(\boldsymbol{w}^{(t+1)}\right)-F\left(\boldsymbol{w}^{(t)}\right)\right\}\leqslant\underbrace{\nabla F\left(\boldsymbol{w}^{(t)}\right)^{\top}\left(\boldsymbol{w}^{(t+1)}-\boldsymbol{w}^{(t)}\right)}_{E_1}+\frac{\rho}{2}\underbrace{\left\|\boldsymbol{w}^{(t+1)}-\boldsymbol{w}^{(t)}\right\|^2}_{E_2}. \tag{D6}$$

Then, we will bound E_1 as follows:

$$\begin{aligned}
E_1&=\frac{1}{N}\sum_{i=1}^{N}\nabla F\left(\boldsymbol{w}^{(t)}\right)^{\top}\left(\boldsymbol{w}_i^{(t+1)}-\boldsymbol{w}^{(t)}\right)=\frac{1}{N}\sum_{i=1}^{N}\nabla F\left(\boldsymbol{w}^{(t)}\right)^{\top}\left(\boldsymbol{w}_i^{(t+1)}-\boldsymbol{w}^{(t)}\right)\\
&=\frac{1}{N}\sum_{i=1}^{N}\nabla F\left(\boldsymbol{w}^{(t)}\right)^{\top}\left(\boldsymbol{w}_i^{(t,\tau)}-\boldsymbol{w}^{(t)}\right)\\
&=\frac{1}{N}\sum_{i=1}^{N}\nabla F\left(\boldsymbol{w}^{(t)}\right)^{\top}\left(\boldsymbol{w}_i^{(t,\tau-1)}-\eta\nabla F_i\left(\boldsymbol{w}_i^{(t,\tau-1)},\widetilde{\mathcal{D}}_i\right)-\boldsymbol{w}^{(t)}\right)\\
&=-\frac{\eta}{N}\sum_{i=1}^{N}\sum_{\ell=0}^{\tau-1}\nabla F\left(\boldsymbol{w}^{(t)}\right)^{\top}\nabla F_i\left(\boldsymbol{w}_i^{(t,\ell)},\widetilde{\mathcal{D}}_i\right).
\end{aligned} \tag{D7}$$

Because $\boldsymbol{a}^{\top}\boldsymbol{b}=\|\boldsymbol{a}\|^2+\|\boldsymbol{b}\|^2-\|\boldsymbol{a}-\boldsymbol{b}\|^2$, we have

$$\begin{aligned}
E_1&=-\frac{\eta}{2N}\sum_{i=1}^{N}\sum_{\ell=0}^{\tau-1}\left(\left\|\nabla F\left(\boldsymbol{w}^{(t)}\right)\right\|^2+\left\|\nabla F_i\left(\boldsymbol{w}_i^{(t,\ell)},\widehat{\mathcal{D}}_i\right)\right\|^2-\left\|\nabla F\left(\boldsymbol{w}^{(t)}\right)-\nabla F_i\left(\boldsymbol{w}_i^{(t,\ell)},\widehat{\mathcal{D}}_i\right)\right\|^2\right)\\
&\leqslant-\frac{\eta}{2N}\sum_{i=1}^{N}\sum_{\ell=0}^{\tau-1}\left(\left\|\nabla F_i\left(\boldsymbol{w}_i^{(t,\ell)},\widehat{\mathcal{D}}_i\right)\right\|^2-\left\|\nabla F\left(\boldsymbol{w}^{(t)}\right)-\nabla F_i\left(\boldsymbol{w}_i^{(t,\ell)},\widehat{\mathcal{D}}_i\right)\right\|^2\right)\\
&\overset{(a)}{\leqslant}-\frac{\eta}{2N}\sum_{i=1}^{N}\sum_{\ell=0}^{\tau-1}\left\|\nabla F_i\left(\boldsymbol{w}_i^{(t,\ell)},\widehat{\mathcal{D}}_i\right)\right\|^2+\frac{\eta}{N}\sum_{i=1}^{N}\sum_{\ell=0}^{\tau-1}\left(\left\|\nabla F_i\left(\boldsymbol{w}_i^{(t,\ell)}\right)-\nabla F_i\left(\boldsymbol{w}_i^{(t,\ell)},\widehat{\mathcal{D}}_i\right)\right\|^2\right)\\
&\quad+\frac{\eta}{N}\sum_{i=1}^{N}\sum_{\ell=0}^{\tau-1}\left(\left\|\nabla F\left(\boldsymbol{w}_i^{(t,\ell)}\right)-\nabla F\left(\boldsymbol{w}^{(t)}\right)\right\|^2\right)\\
&\overset{(b)}{\leqslant}-\frac{\eta}{2N}\sum_{i=1}^{N}\sum_{\ell=0}^{\tau-1}\left\|\nabla F_i\left(\boldsymbol{w}_i^{(t,\ell)},\widehat{\mathcal{D}}_i\right)\right\|^2+\frac{\eta}{N}\sum_{i=1}^{N}\sum_{\ell=0}^{\tau-1}\left(\varepsilon_i+\widetilde{\sigma}_R\sqrt{\frac{(1-\gamma)}{D_{in}}}+\widetilde{\sigma}_S\sqrt{\frac{\gamma}{D_{in}}}\right)\\
&\quad+\frac{\eta\rho^2}{N}\sum_{i=1}^{N}\sum_{\ell=0}^{\tau-1}\left\|\boldsymbol{w}_i^{(t,\ell)}-\boldsymbol{w}^{(t)}\right\|^2,
\end{aligned} \tag{D8}$$

where step (a) applied the triangle inequality, and step (b) uses the Lemma 1 and the ρ-Lipschitz smooth. Due to the gradient descent process, we have

$$\begin{aligned}\left\|\boldsymbol{w}_i^{(t,\ell)}-\boldsymbol{w}^{(t)}\right\|^2&=\left\|\boldsymbol{w}_i^{(t,\ell-1)}-\eta\nabla F_i\left(\boldsymbol{w}_i^{(t,\ell-1)}\right)-\boldsymbol{w}^{(t)}\right\|^2=\eta^2\left\|\sum_{\kappa=0}^{\ell-1}\nabla F_i\left(\boldsymbol{w}_i^{(t,\kappa)}\right)\right\|^2\\&\overset{(a)}{\leqslant}\eta^2(\ell-1)\sum_{\kappa=0}^{\ell-1}\left\|\nabla F_i\left(\boldsymbol{w}_i^{(t,\kappa)}\right)\right\|^2,\end{aligned}\tag{D9}$$

where step (a) is due to $\left\|\sum_{\kappa=0}^{\ell-1}\boldsymbol{a}_\kappa\right\|^2\leqslant(\ell-1)\sum_{\kappa=0}^{\ell-1}\|\boldsymbol{a}_\kappa\|^2$. Substituting (D9) into (D8), we have

$$\begin{aligned}E_1\leqslant&-\frac{\eta}{2N}\sum_{i=1}^{N}\sum_{\ell=0}^{\tau-1}\left\|\nabla F_i\left(\boldsymbol{w}_i^{(t,\ell)},\widehat{\mathcal{D}}_i\right)\right\|^2+\frac{\eta}{N}\sum_{i=1}^{N}\sum_{\ell=0}^{\tau-1}\left(\varepsilon_i+\widetilde{\sigma}_R\sqrt{\frac{(1-\gamma)}{D_{in}}}+\widetilde{\sigma}_S\sqrt{\frac{\gamma}{D_{in}}}\right)^2\\&+\frac{\eta^3\rho^2\tau}{N}\sum_{i=1}^{N}\sum_{\ell=0}^{\tau-1}(\tau-\ell-1)\left\|\nabla F_i\left(\boldsymbol{w}_i^{(t,\ell)}\right)\right\|^2.\end{aligned}\tag{D10}$$

Next, we can bound E_2 as follows:

$$\begin{aligned}E_2=&\mathbb{E}\left\{\left\|\boldsymbol{w}^{(t+1)}-\boldsymbol{w}^{(t)}\right\|^2\right\}=\frac{\eta^2}{N^2}\mathbb{E}\left\{\left\|\sum_{i=1}^{N}\sum_{\ell=0}^{\tau-1}\nabla F_i\left(\boldsymbol{w}_i^{(t,\ell)},\widehat{\mathcal{D}}_i\right)\right\|^2\right\}\\\leqslant&\frac{\tau\eta^2}{N}\sum_{i=1}^{N}\sum_{\ell=0}^{\tau-1}\mathbb{E}\left\{\left\|\nabla F_i\left(\boldsymbol{w}_i^{(t,\ell)},\widehat{\mathcal{D}}_i\right)\right\|^2\right\}\\\leqslant&\frac{\tau\eta^2}{N}\sum_{i=1}^{N}\sum_{\ell=0}^{\tau-1}\mathbb{E}\left\{\left\|\nabla F_i\left(\boldsymbol{w}_i^{(t,\ell)},\widehat{\mathcal{D}}_i\right)-\nabla F\left(\boldsymbol{w}_i^{(t,\ell)}\right)\right\|^2\right\}+\frac{\tau\eta^2}{N}\sum_{i=1}^{N}\sum_{\ell=0}^{\tau-1}\mathbb{E}\left\{\left\|\nabla F\left(\boldsymbol{w}_i^{(t,\ell)}\right)\right\|^2\right\}\\\leqslant&\frac{\tau\eta^2}{N}\sum_{i=1}^{N}\sum_{\ell=0}^{\tau-1}\left(\varepsilon_i+\widetilde{\sigma}_R\sqrt{\frac{(1-\gamma)}{D_{in}}}+\widetilde{\sigma}_S\sqrt{\frac{\gamma}{D_{in}}}\right)^2+\frac{\tau\eta^2}{N}\sum_{i=1}^{N}\sum_{\ell=0}^{\tau-1}\mathbb{E}\left\{\left\|\nabla F\left(\boldsymbol{w}_i^{(t,\ell)}\right)\right\|^2\right\}.\end{aligned}\tag{D11}$$

Substituting (D10) and (D11) into (D6), we have

$$\begin{aligned}\mathbb{E}\left\{F\left(\boldsymbol{w}^{(t+1)}\right)-F\left(\boldsymbol{w}^{(t)}\right)\right\}\leqslant&-\frac{\eta\tau}{2}\sum_{i=1}^{N}\sum_{\ell=0}^{\tau-1}\left(\varepsilon_i+\widetilde{\sigma}_R\sqrt{\frac{(1-\gamma)}{D_{in}}}+\widetilde{\sigma}_S\sqrt{\frac{\gamma}{D_{in}}}\right)^2\\&-\frac{\eta}{2N}\sum_{i=1}^{N}\sum_{\ell=0}^{\tau-1}\mathbb{E}\left\{\left\|\nabla F\left(\boldsymbol{w}_i^{(t,\ell)}\right)\right\|^2\right\}+\eta\tau\sum_{i=1}^{N}\sum_{\ell=0}^{\tau-1}\left(\varepsilon_i+\widetilde{\sigma}_R\sqrt{\frac{(1-\gamma)}{D_{in}}}+\widetilde{\sigma}_S\sqrt{\frac{\gamma}{D_{in}}}\right)^2\\&+\frac{\eta^3\rho^2\tau^2}{N}\sum_{i=1}^{N}\sum_{\ell=0}^{\tau-1}\left\|\nabla F_i\left(\boldsymbol{w}_i^{(t,\ell)}\right)\right\|^2\\&+\frac{\rho\tau^2\eta^2}{2}\sum_{i=1}^{N}\sum_{\ell=0}^{\tau-1}\left(\varepsilon_i+\widetilde{\sigma}_R\sqrt{\frac{(1-\gamma)}{D_{in}}}+\widetilde{\sigma}_S\sqrt{\frac{\gamma}{D_{in}}}\right)^2+\frac{\rho\tau\eta^2}{2N}\sum_{i=1}^{N}\sum_{\ell=0}^{\tau-1}\mathbb{E}\left\{\left\|\nabla F\left(\boldsymbol{w}_i^{(t,\ell)}\right)\right\|^2\right\}\\\leqslant&\frac{\eta\tau(1+\rho\eta\tau)}{2}\sum_{i=1}^{N}\sum_{\ell=0}^{\tau-1}\left(\varepsilon_i+\widetilde{\sigma}_R\sqrt{\frac{(1-\gamma)}{D_{in}}}+\widetilde{\sigma}_S\sqrt{\frac{\gamma}{D_{in}}}\right)^2+\frac{\eta(\eta^2\rho^2\tau^2-1)}{2N}\sum_{i=1}^{N}\sum_{\ell=0}^{\tau-1}\left\|\nabla F_i\left(\boldsymbol{w}_i^{(t,\ell)}\right)\right\|^2.\end{aligned}\tag{D12}$$

Then, summing both sides of (D12) from $t=0$ to $T-1$ yields,

$$\begin{aligned}&\mathbb{E}\left\{F\left(\boldsymbol{w}^{(T)}\right)-F\left(\boldsymbol{w}^{(0)}\right)\right\}\\&\leqslant\frac{\eta\tau(1+\rho\eta\tau)}{2}\sum_{i=1}^{N}\sum_{\ell=0}^{\tau-1}\left(\varepsilon_i+\widetilde{\sigma}_R\sqrt{\frac{(1-\gamma)}{D_{in}}}+\widetilde{\sigma}_S\sqrt{\frac{\gamma}{D_{in}}}\right)^2+\frac{\eta(\eta^2\rho^2\tau^2-1)}{2NT}\sum_{t=0}^{T-1}\sum_{i=1}^{N}\sum_{\ell=0}^{\tau-1}\left\|\nabla F_i\left(\boldsymbol{w}_i^{(t,\ell)}\right)\right\|^2.\end{aligned}\tag{D13}$$

Finally, rearranging the terms in (D13), we have

$$\begin{aligned}&\frac{1}{NT\tau}\sum_{t=0}^{T-1}\sum_{i=1}^{N}\sum_{\ell=0}^{\tau-1}\left\|\nabla F_i\left(\boldsymbol{w}_i^{(t,\ell)}\right)\right\|^2\\&\leqslant\frac{2\left(F\left(\boldsymbol{w}^{(0)}\right)-F\left(\boldsymbol{w}^{\star}\right)\right)}{\eta(1-\eta^2\rho^2\tau^2)}+\frac{\tau(1+\rho\eta\tau)}{1-\eta^2\rho^2\tau^2}\sum_{i=1}^{N}\sum_{\ell=0}^{\tau-1}\left(\varepsilon_i+\widetilde{\sigma}_R\sqrt{\frac{(1-\gamma)}{D_{in}}}+\widetilde{\sigma}_S\sqrt{\frac{\gamma}{D_{in}}}\right)^2.\end{aligned}\tag{D14}$$

From this theorem, we can have the following remarks.

Remark 2. If the number of training samples is larger, i.e., a larger D_{in}, the convergence bound will be smaller. In other words, increasing data samples can improve the FL training performance.

Remark 3. We assume the bounded variance caused by real samples is larger than fake ones, i.e., $\widetilde{\sigma}_S\geqslant\widetilde{\sigma}_R$. When the system adopts more generated samples in the FL training, i.e., a larger γ, the convergence bound will be larger.

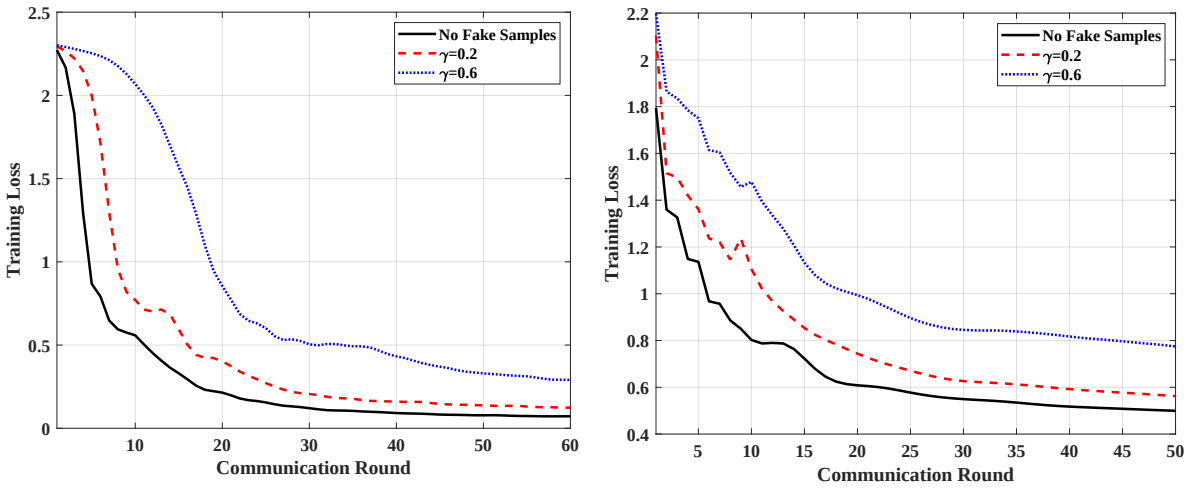

Figure E1 Training loss v.s. Communication rounds with various fake sample ratios. (Left) MNIST Dataset; (Right) Fashion-MNIST Dataset.

Appendix E Experimental Results

Appendix E.1 Datasets

In our experimental results, we use four benchmark datasets for different ML tasks:

- MNIST and Fashion-MNIST dataset, which both have $70,000$ digit images of size 28×28, are split into $60,000$ training and $10,000$ test samples;
- CIFAR-10 dataset, which consists of $60,000$ color images in 10 object classes such as deer, airplane, and dog with 6000 images included per class, is split into $50,000$ training and $10,000$ test samples;
- Purchase dataset [4], which has around $32,000$ tabular samples and each sample has 600 attributes, is split into $20,000$ training and $12,000$ test samples.

We consider the data distribution as independent identically distributed (IID), i.e. clients in the FL system possess the same amount of data from training sets randomly and independently.

Appendix E.2 Parameter and Hardware Settings

We use the multi-layer perceptron (MLP) with 512 neurons in a hidden layer with ReLU unites and softmax for the MNIST and Purchase dataset, and a convolutional neural network (CNN) consists of three 3×3 convolutional layers (the first with 64 filters, the second with 128 filters, the third with 256 filters, each followed with 2×2 max pooling and ReLu activation) and two fully connected layers (the first with 128 units, the second with 256 units, each followed with ReLu activation), and a final softmax output layer for the Fashion-MNIST and CIFAR-10 dataset, as the training model to construct the FL system.

Each client locally computes stochastic gradient descent (SGD) updates on each dataset and then aggregates updates to train a global model. In addition, we set the total number of clients to $M = 50$, and in each communication round, 30 of the 50 clients will be randomly chosen to upload the compressed models. In the experiments, the number of local iterations is set to 5 with a learning rate of 0.02, and the training ends after 80 communication rounds. To record the average results, we run 20 times for each experiment. All the experiments are conducted on computers with 11th Gen Inter(R) Core(TM) i7-1180H @2.30GHZ CPU and NVIDIA GeForce RTX 3070 GPU.

Appendix E.3 Training Loss

In Fig. E1, we provide the training loss of the proposed algorithm under different fake ratios. As can be found in this figure, the proposed algorithm can converge at a similar rate to the one without fake samples. In addition, a smaller fake ratio (γ) has a lower training loss, which is consistent with Remark 3 in the main file.

Appendix E.4 Comparisons on the Successful Rate of the Membership Inference Attack

In this subsection, we provide the performance of the membership inference attack with different fake ratios and various training epochs of the conditional GAN. As can be found in these Table E1 and E2, a larger fake ratio and training epochs will lead to a better performance to against MIA, and it shows the effectiveness of the proposed algorithm.

We have also conducted experimental results to compare the proposed algorithms and baselines (DP-FL) on four datasets as shown in Tab. E3. We can see that the proposed algorithms can achieve a higher test accuracy and a success rate of the membership inference attack compared with baselines in most cases. In addition, we can observe the ratio of fake samples γ is a key factor to balance the privacy risk and training performance.

Table E1 Comparisons of the membership inference attack accuracy with various fake ratios and conditional-GAN training epochs using the MNIST dataset.

Fake ratios	Epochs = 10	Epochs = 50	Epochs = 100
γ =0.6	**0.5112**	**0.5103**	**0.51**
γ =0.2	0.5210	0.5192	0.5178
0.0 (No Fake Samples)	0.584	0.584	0.584

Table E2 Comparisons of the membership inference attack accuracy with various fake ratios and conditional-GAN training epochs the FashionMNIST dataset.

Fake ratios	Epochs = 100	Epochs = 200	Epochs = 400
γ =0.6	**0.528**	**0.5162**	**0.5152**
γ =0.2	0.5364	0.5284	0.5238
0.0 (No Fake Samples)	0.613	0.613	0.613

Table E3 Comparisons of test accuracy and precision of the membership inference attack (the left is the MIA successful rate and the right is the test accuracy).

Dataset	No Privacy	DP-FL ($\epsilon = 4$)	DP-GenFL ($\gamma = 1.0$)	DP-GenFL ($\gamma = 0.6$)	DP-GenFL ($\gamma = 0.2$)
MNIST	0.584/0.977	0.533/0.934	0.510/0.501	0.515/0.947	0.519/0.962
Fashion-MNIST	0.613/0.826	0.543/0.769	0.510/0.550	0.518/0.750	0.533/0.789
CIFAR-10	0.668/0.553	0.543/0.350	0.512/0.212	0.548/0.506	0.551/0.545
Purchase	0.680/0.602	0.550/0.281	0.506/0.200	0.549/0.478	0.598/0.534

Appendix E.5 Related Works

In order to further preserve clients' privacy in FL, many works focus on information stealing or data perturbation to explore possible risks and privacy gains. Via GAN techniques, a designed attack generates prototypical samples of the targeted training set that was meant to be private in the collaborative deep learning framework [12]. Further, the work in [13] leveraged generative adversarial networks (GANs) to affect the learning process by unloading injected model parameters and striving to reconstruct the private data of users by learning hidden features from shared local model parameters. The work in [14] proposed a personalized FL method based on GANs that allows each client to design its own model to participate in FL independently relying on combined datasets (both its local dataset and the generated dataset from other clients). We can note that clients in FL can generate other clients' training samples by uploading some crafted local updates, which brings in additive privacy risks.

Other works aim to prevent clients' privacy from stealing by leveraging differentially private data publication techniques and differentially private GANs. As one of the most popular privacy-preserving data collection mechanisms, local differential privacy (LDP), perturbs each user's data locally and only sends the noisy version of her information to the aggregator [15–17]. The work in [18] proposed efficient multi-dimensional joint distribution estimation algorithms with LDP and then developed a local differentially private high-dimensional data publication algorithm (LoPub) by generating an approximation of the original crowdsourced data based on this distribution estimation technique. The work in [19] studied the utility difference between uploading differentially private GANs (data generator) and differentially privately trained models (analytical algorithms) and then found that the latter is superior to the former in terms of data utility, at the cost of the flexibility in choosing arbitrary analytical algorithms.

We can observe that image data is too sensitive, in which some private features are easy to recognize via generated data based on differentially private GANs. However, existing local differentially private data publication techniques are more suitable for tabular data instead of image data, which limits its application. In addition, it is also an unexplored area to train FL models by synthesizing data using GAN techniques for image data.

References

1 Dwork C, Roth A. The algorithmic foundations of differential privacy. Found. Trends Theor. Comput. Sci., 2014, 9(3-4): 211-407.
2 Mironov I. Rényi differential privacy. In: Proceedings of the IEEE 30th computer security foundations symposium (CSF), 2017: 263-275.
3 Wei K, Li J, Ding M, et al. Federated learning with differential privacy: Algorithms and performance analysis. IEEE Transactions on Information Forensics and Security, 2020, 15: 3454-3469.
4 Shokri R, Stronati M, Song C, et al. Membership inference attacks against machine learning models. In: Proceedings of the IEEE symposium on security and privacy (SP). 2017: 3-18.
5 Rahman M A, Rahman T, Laganie R, et al. Membership Inference Attack against Differentially Private Deep Learning Model. Transactions on Data Pruvacy, 2018, 11(1): 61-79.
6 Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images. Master's thesis, University of Toronto, 2009.

7 Kang Wei et al. User-Level Privacy-Preserving Federated Learning: Analysis and Performance Optimization. IEEE Transactions on Mobile Computing, 2022, 21(9): 3388-3401.

8 Mironov I, Talwar K, Zhang L. Rényi Differential Privacy of the Sampled Gaussian Mechanism. arXiv preprint arXiv:1908.10530, 2019.

9 Balle B, Barthe G, Gaboardi M, et al. Hypothesis testing interpretations and Rényi differential privacy. In: Proceedings of the International Conference on Artificial Intelligence and Statistics. PMLR, 2020: 2496-2506.

10 Lin Z, Sekar V, Fanti G. On the privacy properties of gan-generated samples. In: Proceedings of the International Conference on Artificial Intelligence and Statistics. PMLR, 2021: 1522-1530.

11 Zhang P, Liu Q, Zhou D, et al. On the discrimination-generalization tradeoff in GANs. In: Proceedings of the International Conference on Learning Representations, 2018, Poster.

12 Briland Hitaj, Giuseppe Ateniese, Fernando Fernando Pérez-Cruz. Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS), 2017: 603-618.

13 Y. Sun, N. S. T. Chong and H. Ochiai. Information Stealing in Federated Learning Systems Based on Generative Adversarial Networks. In: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2021: 2749-2754.

14 X. Cao, G. Sun, H. Yu and M. Guizani. PerFED-GAN: Personalized Federated Learning via Generative Adversarial Networks. IEEE Internet of Things Journal, 2022.

15 T. Wang, N. Li and S. Jha. Locally Differentially Private Frequent Itemset Mining. In: Proceedings of the IEEE Symposium on Security and Privacy (SP), 2018: 127-143.

16 Q. Ye, H. Hu, X. Meng and H. Zheng. PrivKV: Key-Value Data Collection with Local Differential Privacy. In: Proceedings of the IEEE Symposium on Security and Privacy (SP), 2019: 317-331.

17 J. Duan, Q. Ye and H. Hu. Utility Analysis and Enhancement of LDP Mechanisms in High-Dimensional Space. In: Proceedings of the IEEE 38th International Conference on Data Engineering (ICDE), 2022: 407-419.

18 X. Ren et al. LoPub: High-Dimensional Crowdsourced Data Publication With Local Differential Privacy. IEEE Transactions on Information Forensics and Security, 2018, 13(9): 2151-2166.

19 Y. -T. Chen, C. -Y. Hsu, C. -M. Yu, M. Barhamgi and C. Perera. On The Private Data Synthesis Through Deep Generative Models for Data Scarsity of Industrial Internet of Things. IEEE Transactions on Industrial Informatics, 2021.