Computer Vision and Image Understanding 238 (2024) 103882

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

FedER: Federated Learning through Experience Replay and privacy-preserving data synthesis

Matteo Pennisi [a,*,1], Federica Proietto Salanitri [a,1], Giovanni Bellitto [a], Bruno Casella [b], Marco Aldinucci [b], Simone Palazzo [a], Concetto Spampinato [a]

[a] Department of Electrical, Electronics and Computer Engineering, University of Catania, 95125, Italy
[b] Department of Computer Science, University of Turin, Turin, 10149, Italy

ARTICLE INFO

Communicated by Nikos Komodakis

MSC:
41A05
41A10
65D05
65D17

Keywords:
Decentralized learning
Federated learning
Privacy in machine learning
Pattern recognition and classification

ABSTRACT

In the medical field, multi-center collaborations are often sought to yield more generalizable findings by leveraging the heterogeneity of patient and clinical data. However, recent privacy regulations hinder the possibility to share data, and consequently, to come up with machine learning-based solutions that support diagnosis and prognosis. Federated learning (FL) aims at sidestepping this limitation by bringing AI-based solutions to data owners and only sharing local AI models, or parts thereof, that need then to be aggregated. However, most of the existing federated learning solutions are still at their infancy and show several shortcomings, from the lack of a reliable and effective aggregation scheme able to retain the knowledge learned locally to weak privacy preservation as real data may be reconstructed from model updates. Furthermore, the majority of these approaches, especially those dealing with medical data, relies on a centralized distributed learning strategy that poses robustness, scalability and trust issues. In this paper we present a federated learning strategy, *FedER*, that, exploiting experience replay and generative adversarial concepts, effectively integrates features from local nodes, providing models able to generalize across multiple datasets while maintaining privacy. FedER is tested on two tasks — tuberculosis and melanoma classification — using multiple datasets in order to simulate realistic *non-i.i.d.* medical data scenarios. Results show that our approach achieves performance comparable to standard (non-federated) learning and significantly outperforms state-of-the-art federated methods. Remarkably, we also observe that FedER enables any node model to be used as a global federation model. Indeed, the experience replay strategy with privacy-preserving synthetic data allows all node models to converge to reach the same optimum without the need of a single shared model. Code is available at https://github.com/perceivelab/FedER.

1. Introduction

Recent advances of deep learning in the medical imaging domain have shown that, while data-driven approaches represent a powerful and promising tool for supporting physicians' decisions, the availability of large-scale datasets plays a key role in the effectiveness and reliability of the resulting models (Irvin et al., 2019; Wang et al., 2017; Cohen et al., 2020). However, the curation of large medical imaging datasets is a complex task: data collection at single institutions is relatively slow and the integration of historical data may require significant efforts to deal with different data formats, storage modalities and acquisition devices; moreover, medical institutions are often reluctant to share their own data, due to privacy concerns. As a consequence, this affects the quality, reliability and generalizability of models trained on local datasets, which unavoidably suffer from bias and overfitting issues, reducing the ability to address future data distribution shifts (Zech et al., 2018). In order to overcome the lack of large-scale datasets, methodological solutions can be adopted: in particular, federated learning (Yang et al., 2019) encompasses a family of strategies for distributed training over multiple nodes, each with its own private dataset, which typically communicate with a central node by sending local model updates, used to train the main model. In this scenario, no data is explicitly shared between nodes, thus addressing the required privacy issues. However, this family of techniques generally performs well when dataset distributions are approximately *i.i.d.* and local gradients/models contribute to learning shared features: unfortunately, in practice this hypothesis rarely holds, due to differences in the acquisition and in the clinical nature of data collected by multiple institutions. Moreover, the presence of a central node, besides representing a single point of failure, requires that all nodes

* Corresponding author.
E-mail address: matteo.pennisi@phd.unict.it (M. Pennisi).
[1] Equal contribution.

https://doi.org/10.1016/j.cviu.2023.103882
Received 22 February 2023; Received in revised form 19 October 2023; Accepted 4 November 2023
Available online 7 November 2023

trust it to correctly and fairly treat updates from all sources: indeed, privacy issues arise when transferring local updates to the "semi-honest" central node (Evans et al., 2018), which might attempt to reconstruct original inputs from gradients or parameter variations (Zhu et al., 2019; Geiping et al., 2020; Zhao et al., 2020). To address the above limitations, we present *FedER*, a federated learning approach that, leveraging experience replay from continual learning (Ratcliff, 1990; Robins, 1995; Rolnick et al., 2019; Buzzega et al., 2020) and generative models (Goodfellow et al., 2014; Mirza and Osindero, 2014; Karras et al., 2020a), proposes a principled way for training local models that approximately converge to the same decisions, without the need of a shared model architecture and of central coordination. *FedER* also enforces privacy preservation through the transmission of synthetic data generated in a way to obfuscate real data patterns.

Specifically, FedER's learning strategy envisages multiple nodes that initially train their local models and a GAN on their own datasets. The GAN will be used in order to generate a privacy-preserving synthesized version of the dataset (buffer). Once local training is completed in a node, its model and the "buffer" of generated synthetic data are sent to a random node of the network. The receiving node then adapts the incoming model using its own data and the received buffer data, in order to limit model's forgetting. Data privacy is ensured through a privacy-preserving generative adversarial network (GAN) that employs a specific loss designed to maximize the distance from real data, while keeping a high level of realism and — as importantly — clinically-consistent features, in order to allow models to be trained effectively.

FedER is tested on two tasks, simulating a *non-i.i.d.* medical scenario: (1) classification of tuberculosis from X-ray data, using Montgomery County and Shenzhen Hospital datasets (Candemir et al., 2013; Jaeger et al., 2014, 2013), and (2) melanoma classification using skin images of the ISIC 2019 dataset (Combalia et al., 2019; Tschandl et al., 2018; Codella et al., 2018). The experimental setting is specifically designed to emulate a realistic medical *non-i.i.d.* scenario, where each node in the federation uses its own dataset. This is in stark contrast with common procedures where non-*i.i.d.* distributions are simulated by splitting a single source dataset. Results show how our approach is able to reach performance similar to using centralized training on all real data together in a single node, while outperforming current state-of-the-art methods, such as FedAvg (McMahan et al., 2017), FedProx (Li et al., 2020) and FedBN (Li et al., 2021). Privacy-preserving capabilities are measured quantitatively by evaluating LPIPS distance (Zhang et al., 2018) between real images and samples generated, respectively, through latent space optimization on a standard GAN and by the proposed approach. Qualitatively, we also show several examples of generated images with corresponding closest match in the real dataset, demonstrating significant differences that prevent tracing back to the original real distribution.

In summary, the overall contributions of the proposed work are the following:

- We propose a decentralized federated learning strategy, based on continual learning principles, designed for medical imaging data, which outperforms server-based federated learning approaches and yields performance similar to standard (non-federated) training settings. Furthermore, experience replay allows local node models to converge to the same decisions, thus making the whole approach behave similarly to server-based aggregation models.
- We propose a GAN-based privacy-preserving mechanism that supports synthetic data sharing through a GAN-based technique designed to minimize patient information leak. This is different from most privacy-preserving techniques based on differential privacy, which degrades performance due to added noise.
- Most approaches for model aggregation in federated learning employ gradient/parameter averaging. These solutions completely neglect any similarity or dissimilarity between merged features, possibly resulting in interference that harm convergence. FedER, instead, takes feature semantics into account when merging models: if a node receives a model that extracts useful features for the local dataset, these can be readily employed and re-used, without the risk of randomly averaging them with other less important features. *FedER*, thus, surpasses the common and straightforward weight/gradient averaging paradigm, replacing it with a principled way for knowledge transfer, which relaxes two of the constraints of the leading federated learning approaches: the presence of a central node and model homogeneity.

2. Related work

Federated Learning (FL) (McMahan et al., 2017) has recently emerged as a family of distributed learning strategies that allow nodes to keep training data private, while supporting the creation of a shared model. In a typical FL setting, a central server sends a model to a set of client nodes; each node fine-tunes the model on its own data, then sends local model updates back to the server; the server aggregates the updates by all nodes into the global model, which is sent back to nodes iteratively until convergence. Given the constraints existing in the medical domain, especially in terms of data sharing, it represents an appropriate test-bench for federated learning methods (Li et al., 2019; Roy et al., 2019; Dayan et al., 2021; Feki et al., 2021). The most straightforward way to aggregate information from multiple nodes is through averaging local models of each client, as proposed in FedAvg (McMahan et al., 2017) and FedProx (Li et al., 2020). However, statistical data heterogeneity is an issue as it may lead to catastrophic forgetting (Kairouz et al., 2021; Goodfellow et al., 2013). FedCurv (Shoham et al., 2019) addresses this limitation by adding a penalty term to the loss driving the local models to a shared optimum. FedMA (Wang et al., 2020) builds a shared global model in a layer-wise manner by matching and averaging hidden elements with similar feature extraction signatures. Our method differs from existing feature integration approaches in that, instead of averaging model updates or gradients, which can be subject to input reconstruction attacks (Geiping et al., 2020; Xie et al., 2018; Zhu et al., 2019), each node attempts to learn features that perform well on its own dataset while retaining knowledge from other nodes, in a more principled way than parameter averaging. The strategy of fitting the global model to local data is also sought by the recent federated *personalized methods*. FedBN (Li et al., 2021), for instance, keeps batch normalization layers private, while other model parameters are aggregated by the central node.

However, the presence of a central node that aggregates local updates simplifies the communication protocol when the number of clients is very large (thousands or millions), but introduces several downsides: it represents a single point of failure; it can become a bottleneck when the number of clients increases (Lian et al., 2017); in general, it may not always be available or desirable in collaborative learning scenarios (Kairouz et al., 2021). In this paper, we deal with *decentralized federated learning*, in which the central node is replaced by peer-to-peer communication between clients: there is no longer a global shared model as in standard FL, but the communication protocol is designed so that all local models approximately converge to the same solution. Decentralized learning is particularly suitable to application in the medical domain, where the number of nodes (i.e., institutions) is relatively low; however, research is still ongoing, and no effective solutions have been established. In Lalitha et al. (2019), a Bayesian approach is proposed to learn a shared model over a graph of nodes, by aggregating information from local data with the model of each node's one-hop neighbors. A secure weight averaging algorithm is proposed in Wink and Nochta (2021), where model parameters are not shared between nodes, but all converge to the same numerical values (with the disadvantages associated to parameter averaging with *non-i.i.d.* data distributions). Other approaches implement different communication strategies based on parameter sharing (e.g., decentralized variants on

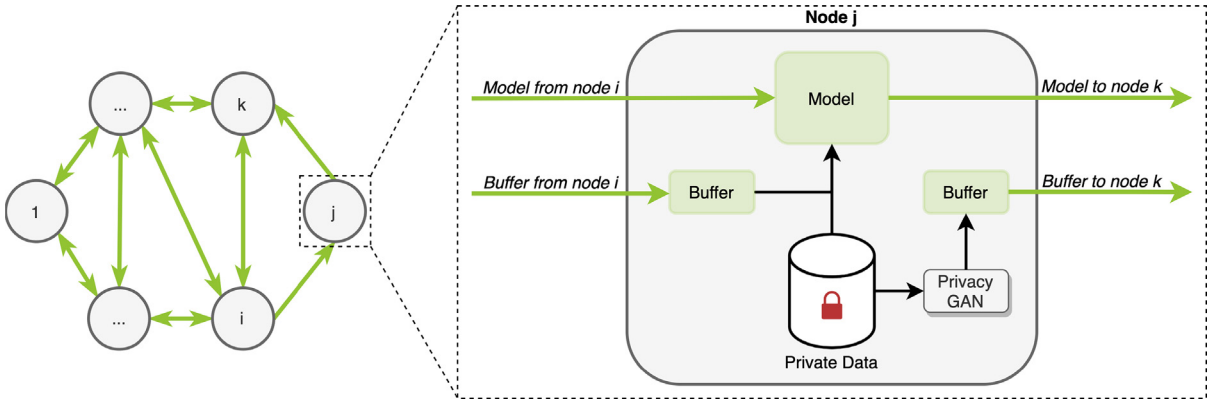

Fig. 1. Overview of FedER learning strategy. Each node initially trains a *privacy-preserving GAN*, that is used to sample synthetic data from the local distribution, without retaining features that may be used to identify patients. Then, each node iteratively receives the local model and a buffer of synthetic samples from a random node, and fine-tunes the received model on its own private data, using the buffer to prevent forgetting of previously-learned features.

FedAvg (Sun et al., 2021; McMahan et al., 2017)). In general, many of the existing solutions do not target, nor are they tested on, the medical domains — most employ toy datasets, such as MNIST and CIFAR10. Two works, similar in the decentralized learning spirit to ours, are proposed in Roy et al. (2019) and Gao et al. (2022), where use cases of decentralized and swarm learning for medical image segmentation are presented. However, like other approaches, they adopt simple parameter averaging to integrate features or predictions from multiple nodes.

3. Method

3.1. Overview

An overview of FedER is shown in Fig. 1. In this scenario, a *federation* consists of a set of N peer nodes, each owning a private dataset.

Before the decentralized training algorithm is started, each node internally trains a *privacy-preserving generative adversarial network*, which is used to generate synthetic samples from its private data distribution. The training objective of the GAN is designed to enforce the constraint that sampled data do not include privacy-sensitive information, while maintaining the clinical features required for successful training.

At each round of decentralized training, each node receives a model and a set of synthetic samples — "buffer" — from a random node in the federation. The input model to the node is fine-tuned on both the private dataset and the buffer, in a way that is reminiscent of experience replay techniques in continual learning (e.g., Buzzega et al., 2020), in order to learn features that transfer between nodes and that can handle non-*i.i.d.* distributions. At the end of each round (i.e., after performing several training iterations), the locally-trained model is sent to a randomly-chosen successor node together with a buffer of local synthetic samples, and the whole procedure is repeated.

In this work we specifically address the problem of federated learning for medical image classification; thus, the method is presented by considering this task, but the whole strategy can be applied to any other task without losing generalization.

3.2. Privacy-preserving GAN

In the proposed method, nodes exchange both models and data, implementing a knowledge transfer procedure based on experience replay (see Section 3.3 below). Of course, sharing real samples would go against federated learning policies; hence, exchanged samples are generated so that they are representative of the local data, while taking precautions against privacy violations — which may happen, for instance, if the generative model overfits the source dataset.

Formally, we assume that each node n_i, from a set of N nodes, owns a private dataset $\mathcal{D}_i = \{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \dots, (\mathbf{x}_M, \mathbf{y}_M)\}$, where each $\mathbf{x}_j \in \mathcal{X}$ represents a sample in the dataset, and each $\mathbf{y}_j \in \mathcal{Y}$ represents the corresponding target.[2] The local dataset can then be used to train a conditional GAN (Mirza and Osindero, 2014), consisting of a generator G, that synthesizes samples for a given label by modeling $P(\mathbf{x}|\mathbf{y}, \mathbf{z})$, where $\mathbf{z} \in \mathcal{Z}$ is a random vector sampled from the generation latent space, and a discriminator D, which outputs the probability of an input sample being real, modeling $P(\text{real}|\mathbf{x}, \mathbf{y})$. The standard GAN formulation introduces a discrimination loss, which trains D to distinguish between real and synthetic samples:

$$\mathcal{L}_D = -\mathbb{E}_{\mathbf{x},\mathbf{y}}\left[\log\left(D(\mathbf{x}, \mathbf{y})\right)\right] - \mathbb{E}_{\mathbf{z},\mathbf{y}}\left[\log\left(1 - D(G(\mathbf{z}, \mathbf{y}), \mathbf{y})\right)\right], \tag{1}$$

and a generation loss, which trains G to synthesize samples that appear realistic to the discriminator:

$$\mathcal{L}_G = -\mathbb{E}_{\mathbf{z},\mathbf{y}}\left[D(G(\mathbf{z}, \mathbf{y}), \mathbf{y})\right]. \tag{2}$$

While it has been theoretically proven that, at convergence, the distribution learned by the generator matches and generalizes from the original data distribution (Goodfellow et al., 2013), unfortunately GAN architectures may be subject to training anomalies, including mode collapse and overfitting: as a consequence, the basic GAN formulation may lead to the generation of samples that are near duplicates of the original samples, which would be unacceptable in a federated learning scenario.

In order to mitigate this risk, we introduce a *privacy-preserving loss*, enforcing the generation of samples that do not retain potentially sensitive information, but still include features that are clinically relevant to the target $\mathbf{y}$ of the synthetic sample. In other words, if $\mathbf{y}$ encodes generic features for the diagnosis of a certain disease, we want the generator to learn how to synthesize samples conditioned by $\mathbf{y}$, that exhibit evidence of that disease but cannot be traced back to any of the dataset's samples of the same disease.

To do so, our privacy-preserving loss aims at penalizing the model proportionally to the similarity between pairs of real and synthetic samples. We measure "similarity" by means of the LPIPS metric (Zhang et al., 2018), which has been shown to capture perceptual similarity by calibrating the distance between feature vectors extracted from a pre-trained VGG model (Simonyan and Zisserman, 2015).

In practice, given a batch of real samples $\left\{\mathbf{x}_1^{(r)}, \mathbf{x}_2^{(r)}, \dots, \mathbf{x}_b^{(r)}\right\}$ and a batch of synthetic samples $\left\{\mathbf{x}_1^{(s)}, \mathbf{x}_2^{(s)}, \dots, \mathbf{x}_b^{(s)}\right\}$, the privacy-preserving

[2] The proposed approach is task-agnostic, as long as it is possible to sample from the $\mathcal{Y}$ distribution. For simplicity, within the scope of this work, we will focus on classification tasks, and we will assume that targets are class labels.

loss term is computed as:

$$\mathcal{L}_{\text{PP}} = \frac{1}{b} \sum_{\mathbf{x}^{(r)}} \sum_{\mathbf{x}^{(s)}} d_L \left(\mathbf{x}^{(r)}, \mathbf{x}^{(s)} \right), \tag{3}$$

where d_L is the LPIPS distance defined as:

$$d_{\text{L}}(\mathbf{x}^{(r)}, \mathbf{x}^{(s)}) = \sum_i w_i \cdot \|\phi_i(\mathbf{x}^{(r)}) - \phi_i(\mathbf{x}^{(s)})\|_2 \tag{4}$$

where ϕ_i represents the feature maps extracted from the ith layer of a deep neural network and w_i is a weight learned to reflect the perceptual importance of that layer. Note that, in this formulation, we ignore the $\mathbf{y}$ targets associated to each $\mathbf{x}$: we want to prevent the model from generating near-duplicates of real samples in general, regardless of class correspondence. Also, we intentionally employ a pairwise metric on samples, rather than an aggregated metric such as Frêchet Inception Distance (Heusel et al., 2017), since we want to prevent similarity between samples, not between distributions, which would conflict with the GAN objective.

The resulting new loss for the Generator is a combination of Eqs. (2) and (3):

$$\mathcal{L}_{G\text{-PP}} = \mathcal{L}_G - \alpha \mathcal{L}_{\text{PP}} \tag{5}$$

where $\mathcal{L}_{\text{PP}}$ is sign reversed as we want to maximize Eq. (3), while α is a hyperparameter used to balance the two terms.

The combined effect of the three loss terms — $\mathcal{L}_D$, $\mathcal{L}_G$, $\mathcal{L}_{\text{PP}}$ — pushes the generator to explore the sample space to match the dataset distribution, while "avoiding" latent space mappings that would project to actual real samples.

3.3. Federated learning with experience replay

Current approaches for federated learning are mostly based on parameter averaging (e.g., FedAvg), which is, however, a straightforward way to combine knowledge from multiple sources: feature locations are not aligned over different models and may be disrupted by updates, before slowly converging to consensus: hypothetically, two models could learn the same set of features at different locations of the same layer, to only have them cancel each other when averaging. In a decentralized scenario, this issue is even exacerbated, due to the lack of an entity that enforces global agreement on node features.

In our approach, we address this problem by taking inspiration from continual learning strategies (Delange et al., 2021) that learn how to perform a task with a non-*i.i.d* data stream without forgetting previously-learned knowledge: as a consequence, models are encouraged to reuse and adapt features so that they can equally serve the current and previous tasks. Analogously, in the federated learning setting, the objective is to train a global model trained on disjoint non-*i.i.d.* data distributions coming from different nodes.

Given these premises, we define a federated learning strategy where a node receives another node's model and surrogate data (generated through our privacy-preserving GAN) — the *"previous task"* — and fine-tunes that model on its own private date — the *"current task"* — while using received synthetic data as a reference to what is necessary to retain/adapt from the knowledge learned by the previous node. The idea is to build for each node a model able to tackle its internal data while not forgetting about the data seen in previous nodes/iterations.

We first introduce the terminology used in the method's description. In our approach, we define a *set of N tasks* $\mathcal{T} = (T_1, T_2, \dots, T_N)$, where T_i is the task to be solved within node n_i.

Definition 1. Task T_i aims at optimizing a model M_i, parameterized by θ_i, on dataset $\mathcal{D}_i$ residing on node n_i and that cannot be shared to other nodes.

Definition 2. A buffer $\mathcal{B}_i$ is a set of synthetic images, drawn from a latent space learned through a generative model $\mathcal{G}_i$ using data $\mathcal{D}_i$ available on node n_i.

Definition 3. Training is organized in parallel *rounds*. At the end of round r, each node n_i produces a model M_i^r trained on dataset $\mathcal{D}_i$ and on a buffer $\mathcal{B}_j$, received from another node n_j, to optimize an objective $\mathcal{L}$, i.e., to find $\arg\min_{\theta_i^r} = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}_i\cup\mathcal{B}_j}[\mathcal{L}(M_i^r(\mathbf{x}, \theta_i^r), \mathbf{y})]$. For each training round, all nodes in parallel share to/receive from other nodes, buffer of synthetic images and trained models.

In the following, we describe our method (whose graphical representation is given in Fig. 1) from the point of view of a single node n_j. At a given round r, training for node n_j can be seen as learning a new task T_j, from dataset $\mathcal{D}_j$, in a continual learning setting by finetuning the incoming model $\mathcal{M}_i^{r-1}$ (with parameters θ_i^{r-1}) on $\mathcal{D}_j$ and on the incoming buffer $\mathcal{B}_i$ in order to learn T_j while mitigating the forgetting of T_i. Thus, unlike other federated learning approaches, each node does not have its own local model: as the decentralized learning strategy proceeds, a node iteratively receives a model from another node and updates it with local information, while preserving previously-learned knowledge, before sending it to the next node. Formally, the loss function for model $\mathcal{M}_j^r$ in node n_j at round r is given as:

$$\begin{aligned}\mathcal{L}(\theta_j^r) = {} & \lambda \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}_j}[\mathcal{L}(M_j^r(\mathbf{x}, \theta_j^r), \mathbf{y})] \\ & + (1-\lambda)\mathbb{E}_{(\mathbf{x}',\mathbf{y}')\sim\mathcal{B}_i}[\mathcal{L}(M_j^r(\mathbf{x}', \theta_j^r), \mathbf{y}')]\end{aligned} \tag{6}$$

where λ controls the importance between real samples from the local dataset $\mathcal{D}_i$ and replayed synthetic samples from node n_i. Note that, for a given n_j, the predecessor node n_i is not fixed: in a practical asynchronous implementation, a node may receive a model and buffer from any random node in the federation at any time, using queues to handle incoming data.

After optimizing the $\mathcal{L}(\theta_j^r)$ objective through mini-batch gradient descent for a certain number of training iterations, the resulting model $M_j^r(\theta_j^r)$, with updated parameters θ_j^r, is sent to a random node n_k of the federation, along with a buffer $\mathcal{B}_j$ of locally-generated synthetic samples. The number of training rounds/iterations and the size of the buffer is discussed in the next section.

Then, the general federated model $\mathcal{M}$, after all training rounds, is given by the union of all the N node models, i.e., $\mathcal{M} = M_1 \cup M_2 \cup \dots \cup M_N$. However, experimental results, reported below in Section 4, demonstrate that all models converge to similar decisions, thus each node model can be considered as a general model for the entire network.

To ease the understanding of the whole training strategy we also report the algorithm pseudo-code in Alg. 1.

4. Experimental results

We test FedER on two applications simulating real case scenarios with multiple centers holding, and not sharing, their own data: (1) tuberculosis classification from X-ray images using two different datasets, and (2) skin lesion classification with three different datasets. In this section we present the employed benchmarks, the training procedure and report the obtained results to demonstrate the advantages of the proposed approach w.r.t. the state-of-the-art.

4.1. Datasets

X-ray image datasets for tuberculosis classification. We assume two separate nodes in the federation: one with the Montgomery County X-ray set and another one with the Shenzhen Hospital X-ray set (Candemir et al., 2013; Jaeger et al., 2014, 2013). The Montgomery Set consists of 138 frontal chest X-ray images (80 negatives and 58 positives), captured with a Eureka stationary machine (CR) at 4020 × 4892 or 4892 × 4020 pixel resolution. The Shenzhen dataset was collected using a Philips DR Digital Diagnostic system. It includes 662 frontal chest X-ray images (326 negatives and 336 positives), with a variable resolution of approximately 3000 × 3000 pixels.

Algorithm 1: FedER Learning Procedure

Notations *The N nodes are indexed by n_i; E is the number of local epochs for each round. R the total round of communications between nodes.*
Each node n_i contains:
$\mathcal{D}_i$ Private Dataset
$\mathcal{G}_i$ Generator (privacy-preserving) trained on $\mathcal{D}_i$
$\mathcal{M}_i^r$ Model for node n_i at round r
$\mathcal{B}_i$ Synthetic data buffer sampled using G_i

// Before Federated Training
for *each node* $n_i \in N$ **do**
  Train $\mathcal{G}_i$ on $\mathcal{D}_i$
  Generate Buffer $\mathcal{B}_i$ using $\mathcal{G}_i$
  Train $\mathcal{M}_i^0$ on $\mathcal{D}_i$
end

// Federated Training
for *each round* $r = 1, 2, ..., R$ **do**
  for *each node* $n_j \in N$ *in parallel* **do**
    Send $\mathcal{M}_j^{r-1}$, $\mathcal{B}_j$ to a node $n_k \in \{N \setminus n_j\}$
    Receive $\mathcal{M}_i^{r-1}$, $\mathcal{B}_i$ from a node $n_i \in \{N \setminus n_j\}$
    $\mathcal{M}_j^r \leftarrow \mathcal{M}_i^{r-1}$
    Train $\mathcal{M}_j^r$ on $\{\mathcal{D}_j \cup \mathcal{B}_i\}$ for E epochs
  end
end

Skin lesion classification. We employ the ISIC 2019 challenge dataset, which contains 25,331 skin images belonging to nine different diagnostic categories. In this case, we assume a federation with three nodes as data provided belongs to three different sources: (1) the BCN20000 (Combalia et al., 2019) dataset, consisting of 19,424 images of skin lesions captured from 2010 to 2016 in the Hospital Clínic in Barcelona; (2) the HAM10000 dataset (Tschandl et al., 2018), which contains 10,015 skin images collected over a period of 20 years from two different sites, the Department of Dermatology at the Medical University of Vienna, Austria, and the skin cancer practice of Cliff Rosendahl in Queensland, Australia; (3) the MSK4 (Codella et al., 2018) dataset, which is anonymous and includes 819 samples. Among all skin lesion classes, we only consider the melanoma class, posing the problem as a binary classification task.

In all tasks and datasets we adopt 80% of the available data to train both the privacy-preserving GAN and the classification model, while the remaining 20% of each dataset is used as test set. Test sets are also balanced w.r.t. the label to avoid performance biases due to class imbalance. For all tested federated methods (including state-of-the-art ones), model selection is carried out through with 5-fold cross-validation on the training set, as a grid search on number of training rounds, number of rounds per epoch and learning rate. For FedProx (Li et al., 2020), we also include the μ hyperparameter.

4.2. Training procedure and metrics

4.2.1. Federated training

In all settings, we employ ResNet-18 as classification model, trained by minimizing the cross-entropy loss with mini-batch gradient descent using the Adam optimizer. Mini-batch size is set to 32 and 8 for the Shenzhen and Montgomery datasets, respectively, and to 64 for skin lesion datasets. The learning rate was found, through cross-validation, to be 10^{-4}. Data augmentation is carried out with random horizontal flip; for skin images we additionally apply random 90-degree rotations. All images are resized to 256×256. The ratio between real and synthetic samples controlled by λ in Eq. (6) is set to 0.5 for all experiments, i.e., each mini-batch is composed by the same quantity of real and synthetic images. This also ensures that our method performs the same number of optimization steps as other approaches that do not use any synthetic data.

The node federation is trained for R rounds. In our implementation, at each round nodes are randomly ordered to establish each node's predecessor and successor: given our focus on medical applications, we can assume that the number of nodes is low enough that synchronization is not an issue. However, asynchronicity can be achieved by assuming that nodes can store incoming data in a queue: if the distribution of successor nodes is uniform and computation times are similar for all nodes, this is on average equivalent to the synchronous case. The number of rounds R and epochs E for FedER on the tuberculosis and melanoma classification tasks are set both to 100, according the 5-fold cross-validation results shown in Table 1. Buffer size is set for all experiments to 512.

4.2.2. GAN training

We recall that GAN training is carried out **before** federated learning using training data only, while leaving out test samples, as mentioned in Section 4.1. Our privacy-preserving GAN employs StyleGAN2-ADA (Karras et al., 2020b) as a backbone, because of its suitability in low-data regimes and its generation capabilities. Training is carried out in two steps: (1) the GAN is initially trained without any privacy-preserving loss to support learning of high-quality visual features; (2) afterwards, we enable privacy-preserving loss and fine-tune the model in order to limit the embedding of patient-specific patterns in the GAN latent space. For classification purposes, GANs are trained in a label-conditioned fashion with a mini-batch size of 32 and learning rate of 0.0025 for both the generator and the discriminator. Early-stopping criteria are based on the Frêchet Inception Distance (FID) (Heusel et al., 2017) between real and synthetic distributions: in the first training step, we stop training if FID does not improve for 10,000 iterations; in the second training step, we employ a criterion which stops training if FID increases by a factor of 2.5 w.r.t. the value obtained in the first step. As for the α parameter in Eq. (3), we tested multiple values of α (0, 0.5, 1, 1.5, 2 and 3) and found that the value of 1 yields the best compromise between image generation quality and pairwise LPIPS distance (Zhang et al., 2018) over all tested datasets. In order to quantitatively evaluate privacy preservation, we also compute the average LPIPS distance between each real image and its closest synthetic sample by means of latent space projection (described in Section 4.4): the higher value of LPIPS, the lower the possibility to reconstruct real images from the generator.

4.3. Federated learning performance

We first evaluate the performance (in terms of classification accuracy) of FedER in the non-*i.i.d.* setting, and compare it to several centralized baselines, namely:

- **Centralized training**: all datasets are merged in a single node where all training happens. In this setting, no federated learning constraints are applied.
- **Centralized training with synthetic data only**. In this setting, each node trains a privacy-preserving GAN model and shares a synthetic version of its own data with the central node, where global training is performed. In this case, we aim to assess how much information is retained by synthetic data to support classification.
- **Centralized training with synthetic and real data.** This setting is a combination of the previous two: real and synthetic samples are centrally merged and used for training a global classifier. This scenario measures the contribution of synthetic data as a data augmentation approach.

Table 1
Rounds and epochs in FedER. Results (mean ± standard deviation) obtained with 5-fold cross-validation. Buffer size = 512.

Rounds	Epochs	Tuberculosis		Melanoma		
		Shenzhen	Montgomery	BCN	HAM	MSK4
		Accuracy	Accuracy	Accuracy	Accuracy	Accuracy
10	1	82.39 ± 6.91	56.13 ± 3.03	76.73 ± 2.07	82.24 ± 4.01	67.93 ± 4.84
	10	82.86 ± 2.44	86.73 ± 4.22	83.83 ± 1.96	84.72 ± 2.29	73.67 ± 2.59
	100	83.56 ± 1.72	90.79 ± 3.92	85.51 ± 1.85	88.65 ± 1.12	71.81 ± 2.04
100	1	83.31 ± 2.59	88.71 ± 3.82	78.94 ± 2.55	87.34 ± 1.62	72.07 ± 3.45
	10	85.22 ± 2.42	89.72 ± 3.46	84.62 ± 1.40	85.05 ± 1.62	73.72 ± 2.41
	100	87.10 ± 2.31	91.50 ± 2.60	86.06 ± 0.96	89.26 ± 1.11	72.41 ± 1.53

Table 2
Comparison between FedER and centralized baselines. Results for FedER are obtained with a buffer size of 512, 100 rounds and 100 epochs per round.

Methods	Tuberculosis			Melanoma			
	Shenzhen	Montgomery	Mean	BCN	HAM	MSK4	Mean
Standalone	82.31	90.00	86.16	82.90	82.55	69.75	78.40
Centralized training	82.77	77.67	80.22	78.80	82.90	71.23	77.64
Centralized training with synthetic data only	76.92	79.33	78.13	60.71	61.09	61.23	61.01
Centralized training with synthetic data and real data	85.38	86.67	86.03	81.53	80.44	73.46	78.48
FedER (ours)	80.15	86.67	83.41	82.11	84.58	68.40	78.36

Table 3
Accuracy convergence among distributed node models. Each local model is evaluated on all test sets of the federation in order to measure convergence and generalization (lower standard deviation corresponds to higher convergence).

	Dataset	FedER	Standalone
Tuberculosis	Shenzhen	80.54 ± 1.20	66.15 ± 22.84
	Montgomery	85.67 ± 2.36	70.00 ± 28.28
Melanoma	BCN	82.87 ± 1.22	65.06 ± 19.68
	HAM	84.45 ± 0.75	59.94 ± 20.47
	MSK4	67.78 ± 1.28	65.43 ± 5.05

We also compare FedER against standard training of the local node models, referred to as "Standalone". Classification accuracy is computed using local node models on their own data. The results, reported in Table 2, show that standalone training appears to be the most favorable scenario. Centralized strategies perform generally worse than standalone training, because of the non-*i.i.d.* nature of the data. However, when the centralized approach is trained with original data augmented with synthetic samples, its classification accuracy is on par with the standalone training, possibly due to the learned generative latent spaces that likely tend to smooth different modes of non-*i.i.d.* data. FedER, instead, outperforms its centralized counterpart and yields slightly worse performance (1.5 percent points less) than standalone training. Although this may appear, at a first glance, as a shortcoming of FedER, we recall that in a federated learning scenario, we aim at building a model that, leveraging multiple data distributions present in the federation, may generalize better, thus addressing possible future data drifts. In order to assess the capabilities of the trained models to achieve such a generalization, we measure the decision convergence by evaluating how a local node model performs on other node datasets. Results are in Table 3 and show a good average accuracy, with a low standard deviation, by FedER, indicating that each node model performs equally well on its own dataset and on the others (i.e., all node models converge to similar decisions). Conversely, standalone training yields significantly lower accuracy and higher standard deviation than ours, demonstrating to be an unsuitable strategy for the sought generalization properties.

Thus, Table 2 shows the performance obtained by each node model on its internal test data, while Table 3 shows, instead, the performance obtained when each node model is tested again all other nodes' data. The latter results indicate that in FedER, any arbitrary node model can be used for the final evaluation, as all federation models converge to the same decisions. However, we further investigate whether building an ensemble of all node models yields better performance than using one arbitrary model. Results are given in Table 5 indeed showing higher accuracy by the ensemble. However, the models' ensemble leads to increased communication overhead (after training, all models have to be shared across the federation) and inference costs (each node needs to make a forward pass for all its available models to make the prediction). For this reason, the following experiments are carried out without using ensemble.

We then compare our approach (without ensemble) to state-of-the-art federated learning approaches, namely: (a) server-based federated methods, FedAvg (McMahan et al., 2017) and FedProx (Li et al., 2020), which have shown to perform generally better than decentralized methods (Sun et al., 2021; Lalitha et al., 2019), and (b) a personalized method, FedBN (Li et al., 2021). As already mentioned, to avoid biased assessment, we use the official code repository[3] of FedBN (Li et al., 2021) and hyper-parameter selection on the tested datasets was carried out through grid search on training rounds/epochs, learning rate and μ for FedProx (Li et al., 2020) using 5-fold cross validation as for our approach. Results, for the tuberculosis and the melanoma tasks, are reported in Table 4 and show that FedER outperforms all methods under comparison. Interestingly, FedER learning strategy does better than: (a) *server-based methods*, FedAvg (McMahan et al., 2017) and FedProx (Li et al., 2020), suggesting that experience replay is a more effective feature aggregation approach than naive parameter averaging; (b) personalized methods, such as FedBN (Li et al., 2021), which affects a limited aspect of feature representation (i.e., input layer distributions), while our approach adapts the entire model to local and remote tasks.

These above results suggest that experience replay plays a key role in federated models as a principled way to integrate features coming from different data distributions. To further assess its contribution, we evaluate FedER performance when using buffer at different sizes. Results on the tuberculosis task, measured as mean and standard deviation of the local node models over a given dataset, are shown in Table 6 and indicate a clear contribution of the buffer in terms of overall performance and models' agreement. Indeed, with no buffer we obtain the lowest average performance and the highest standard deviation. As the buffer is enabled, we can observe a performance gain (mainly for the Shenzhen dataset) and a significant drop in standard deviation. Performance improves as buffer size increases, although gain becomes negligible above 512. Since higher buffer sizes result in more data to be shared among nodes, we use a buffer size of 512, as the best trade-off between accuracy and communication costs.

[3] https://github.com/med-air/FedBN.

Table 4
Comparison with state-of-the-art methods. In bold, best accuracy values.

	Tuberculosis			Melanoma			
	Shenzhen	Montgomery	Mean	BCN	HAM	MSK4	Mean
FedAvg (McMahan et al., 2017)	72.31	83.33	77.82	77.55	75.15	67.28	73.33
FedProx (Li et al., 2020)	78.46	76.67	77.56	78.80	81.87	64.81	75.16
FedBN (Li et al., 2021)	63.08	70.00	66.54	**82.19**	81.12	59.26	74.19
FedER (ours)	**80.15**	**86.67**	**83.41**	82.11	**84.58**	**68.40**	**78.36**

Table 5
Accuracy performance with and without models' ensemble. Results are computed by testing (first line) each node model with its own data and (second line) creating an ensemble and testing it on all nodes' data.

Method	Tuberculosis	Melanoma
No ensemble	83.41 ± 4.61	78.36 ± 8.72
Ensemble	**84.77 ± 4.57**	**80.35 ± 9.42**

Table 6
FedER classification accuracy w.r.t. buffer size. Each local model is evaluated on all test sets of the federation in order to measure convergence and generalization (lower standard deviation corresponds to higher convergence).

Buffer	Node convergence	
	Shenzhen	Montgomery
0	70.62 ± 11.97	80.33 ± 10.84
256	80.46 ± 2.96	81.67 ± 4.24
512	80.54 ± 1.20	85.67 ± 2.36
1024	82.23 ± 1.31	86.00 ± 3.01
2048	82.08 ± 1.39	88.67 ± 2.97

We finally evaluate the capability of FedER to scale with the size of the federated network. Accordingly, we quantify this property using an *i.i.d.* setting on both tuberculosis (Shenzhen dataset) and skin lesion classification (BCN dataset) tasks, by equally splitting the available data on multiple nodes. Fig. 2 shows how the proposed approach is able to keep classification accuracy high and performs on par with state-of-the-art approaches (namely, FedAvg, FedProx and FedBN).

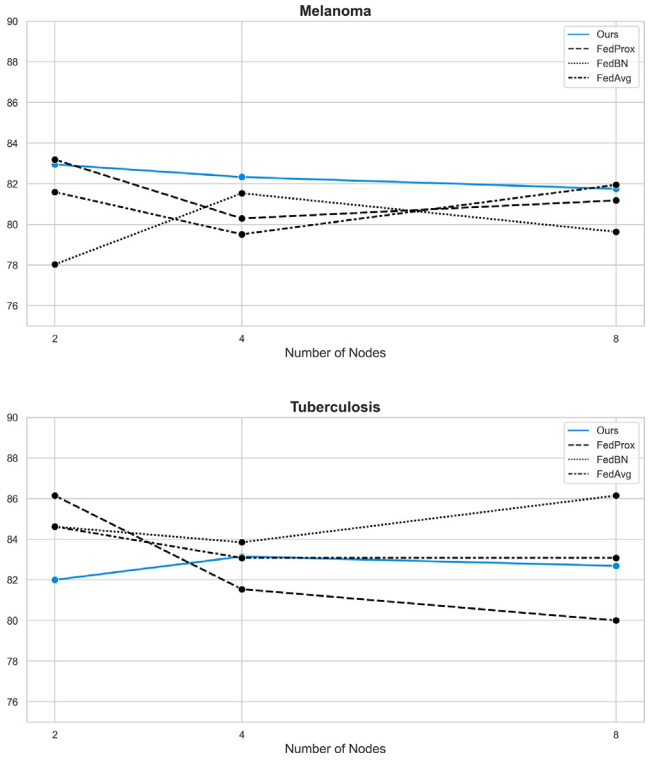

Fig. 2. Scalability performance in the *i.i.d.* setting w.r.t. number of nodes for the proposed approach and state-of-the-art methods.

4.4. Privacy-preserving performance

In this section we quantify how much information of real samples is retained by our privacy-preserving method, and in particular in the mapping between latent space and synthetic images. To do so, we employ the projection method proposed in Karras et al. (2020a): given a real image $\mathbf{x}$, we find an intermediate latent point $\mathbf{w}$ such that the generated image $G(\mathbf{w})$ is most similar to $\mathbf{x}$, by optimizing $\mathbf{w}$ to minimize the LPIPS distance (Zhang et al., 2018) between $\mathbf{x}$ and $G(\mathbf{w})$.

In practice, for each image of the dataset used for GAN training, we perform backprojection to find its most similar synthetic sample, and measure the LPIPS distance between the original and projected images. Fig. 3 shows the histograms of the resulting distances on the Shenzhen dataset, using GAN models trained with and without the proposed privacy-preserving loss (both models start from the same $\mathbf{w}$, for fairness). The histograms show that standard GAN training, with no privacy-preserving loss, tends to yield distances closer to 0, demonstrating that real images are indeed included into the generator latent space; while our model significantly mitigates this issue, by synthesizing samples that are substantially different than the original ones. In order to qualitatively substantiate these findings, Fig. 4 compares original samples from the Shenzhen dataset with the corresponding projections, generated with and without our privacy-preserving loss.[4] It is easy to notice that generated samples with a traditional GAN highly resemble real data, making it impossible to share such samples, albeit synthetic, in a privacy-safe manner, as they clearly contain patient information.

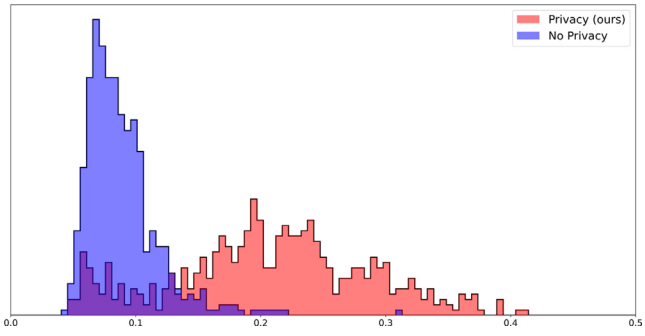

Fig. 3. Quantitative analysis of privacy-preserving generation. In blue, LPIPS distance histogram between real images and the corresponding images obtained through latent space projection using a GAN trained without the proposed privacy-preserving loss. In red, LPIPS distance histogram between real images and the closest images generated with the proposed approach. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Instead, comparing real images with the projections obtained from privacy-preserving GAN confirms the inability of the generator to find latent representations that recover real images used during training.

Given the high realism of generated samples, we run additional tests by proposing two FedER variants aiming to increase the level of privacy preservation: (a) *FedER-A*: models are not shared among nodes

[4] We show only X-ray synthesized samples, as the effect is of our privacy-preserving strategy, is more appreciable than in skin lesion data.

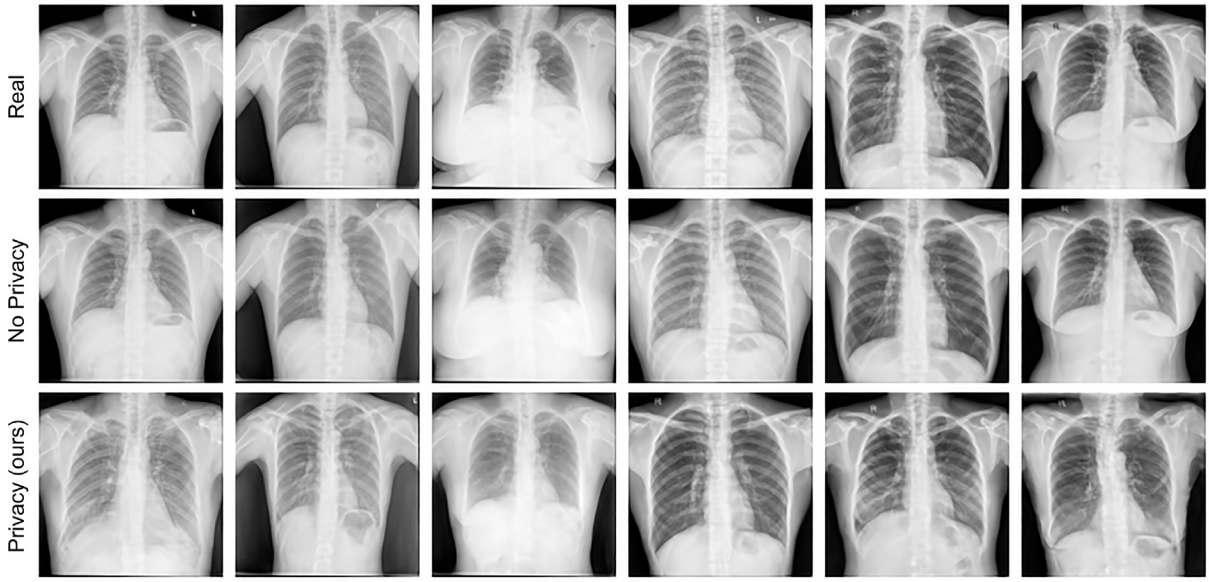

Fig. 4. Qualitative samples of our privacy-preserving generation. Top row: real images from the Shenzhen dataset. Middle row: projection with a standard GAN. Bottom row: projection with our privacy-preserving GAN.

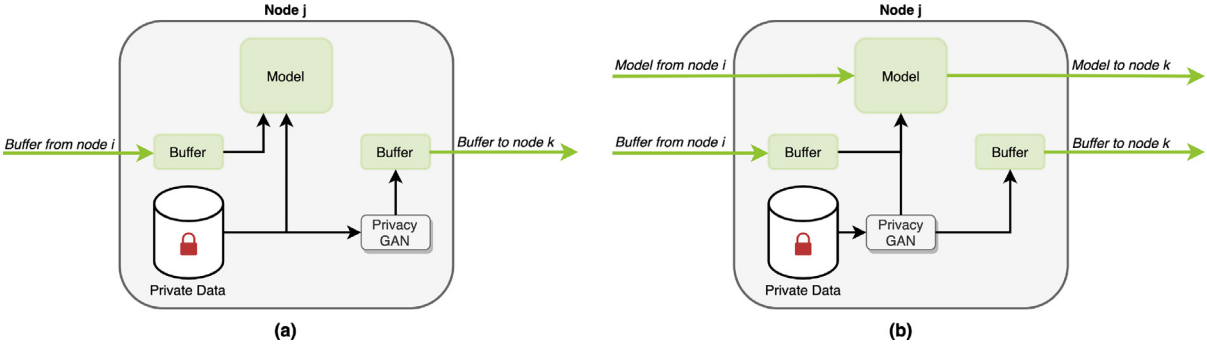

Fig. 5. Privacy-enhanced alternative architectures. *(a) FedER-A* configuration ("Buffer-only sharing"): a local node model is trained on real data, but only a buffer of synthetic samples is shared with other nodes. *(b) FedER-B* ("Synthetic-only training"): Even within the dataset owner node, models are trained on synthetic data only.

— only synthetic buffers are sent and received; (b) *FedER-B*: models are trained only using synthetic data, even on local nodes. Fig. 5 shows the internal architecture of each node in the two variants. Results obtained with these alternative privacy-enhanced configurations are provided in Table 7. It can be noted that FedER-A (i.e., "buffer-only sharing") configuration achieves comparable performance to our standard FedER (82.76 vs. 83.41), but, remarkably, it outperforms all existing federated learning methods on the same datasets (compare Table 4 with the node performance block in Table 7). The FedER-B (i.e., "synthetic-only training") configuration, instead, performs slightly worse than the other two configurations, but is still on par with existing federated methods.

Table 7
Classification accuracy of the proposed privacy-enhanced strategies in the *non-i.i.d.* setting. FedER-A: only buffers are shared (Fig. 5-a). FedER-B: models are trained on synthetic data only (Fig. 5-b). Node performance measures how each node model performs on its own private dataset, while node convergence assesses how a node model performs on other federation nodes.

Config	Node performance			Node convergence	
	Shenzhen	Montgomery	Mean	Shenzhen	Montgomery
FedER	80.15	86.67	83.41	80.54 ± 1.20	85.67 ± 2.36
FedER-A	83.54	82.00	82.76	78.84 ± 6.64	81.00 ±3.30
FedER-B	74.15	81.33	77.74	73.61 ± 4.68	80.40 ± 3.60

4.5. Communication and computational performance

We conclude the experimental analysis by measuring *communication* and *computational* costs.

As for *communication costs*, compared to state-of-the-art approaches, FedER requires additional transmission of synthetic images between nodes at each round. Table 8 reports per-node communication costs for state-of-the-art models (the table reports FedAvg, but the same values apply for FedProx and FedBN) and for FedER, in its full formulation and in the FedER-A variant, where only buffers of synthetic data are shared. The main cost for state-of-the-art models lies in the transfer of the model, and depends on the specific architecture (we included ResNet-18 and ResNet-152 as representative examples of different model scales). Values for our approach are reported for buffers of size 512 containing 256 × 256 images, and depend on the color space. For our full FedER model, the increment in communication costs is significant but not excessive. However, if we take into consideration the variant where only synthetic data are exchanged (i.e., FedER-A), which still performs better than state-of-the-art methods (Tables 4 and 7), communication overhead becomes significantly less than model-sharing approaches.

As for *computational costs* of federated training, FedER incurs the same overhead for parameter optimization and aggregation as state-of-the-art methods. Additionally, before federated training starts, FedER

Table 8
Communication results comparison.

	Tuberculosis		Melanoma	
	ResNet-18	ResNet-152	ResNet-18	ResNet-152
FedAvg FedProx FedBN	45 MB	230 MB	45 MB	230 MB
FedER	65 MB	250 MB	105 MB	290 MB
FedER-A	20 MB	20 MB	60 MB	60 MB

requires that each node trains a local privacy-preserving GAN off-line; this, however, does not affect online federated learning costs, as it is carried out only once at the very beginning of the whole procedure.

Furthermore, we argue that, in the medical domain, the number of institutions in a federation is relatively low and it is reasonable to assume that nodes can benefit from a powerful communication network and computing infrastructure: thus, the overhead introduced by FedER is tolerable, in light of the methodological advantages and the obtained performance and generalization capabilities showed by the resulting models.

5. Conclusion

In this paper, we propose FedER, a decentralized federated learning framework that replaces traditional parameters averaging with a more principled feature integration approach based on the combination of experience replay and privacy-preserving generative models. In FedER, nodes communicate with each other by sharing local models and buffers of synthetic samples; local model updates are carried out in a way that encourages the reuse and adaptation of features learned by other nodes, thus avoiding potentially disruptive effects due to blind feature averaging. Experimental results show that our method outperforms significantly state-of-the-art server-based approaches in a non-*i.i.d.* scenario, which is a typical setting in the medical domain. Additionally, quantitative and qualitative analysis shows that our privacy-preserving generation approach is able to synthesize samples that are significantly different from real data, while correctly supporting the learning of discriminative features. In the future, we aim at investigating some unexplored properties of our method: for instance, unlike all other existing methods based on parameter averaging is required, our approach does not strictly require that all nodes share the same model architecture. Model heterogeneity could therefore be employed to create a shared ensemble and combine different feature learning capabilities.

CRediT authorship contribution statement

Matteo Pennisi: Conceptualization, Methodology, Software, Validation, Writing – original draft, Writing – review. **Federica Proietto Salanitri:** Conceptualization, Methodology, Software, Validation, Writing – original draft, Writing – review. **Giovanni Bellitto:** Conceptualization, Methodology, Software, Writing – original draft, Writing – review. **Bruno Casella:** Software, Writing – original draft, Writing – review. **Marco Aldinucci:** Conceptualization, Resources, Writing – original draft, Writing – review, Supervision. **Simone Palazzo:** Conceptualization, Methodology, Writing – original draft, Writing – review, Supervision. **Concetto Spampinato:** Conceptualization, Methodology, Resources, Writing – original draft, Writing – review, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Code is public, the link to the repo is reported at the end of the abstract. We employed public datasets (references are included in the paper).

Acknowledgments

This research was supported by the following grants: (1) Future Artificial Intelligence Research (FAIR) PNRR MUR Cod. PE0000013- CUP: E63C22001940006; (2) Italian Ministerial grant PRIN 2020 "LEGO.AI: LEarning the Geometry of knOwledge in AI systems", n. 2020TA3K9N, CUP: E63C20011250001.

Matteo Pennisi is a Ph.D. student enrolled in the National PhD in Artificial Intelligence, XXXVII cycle, course on Health and life sciences, organized by Università Campus Bio-Medico di Roma.

References

Buzzega, P., et al., 2020. Dark experience for general continual learning: a strong, simple baseline. In: NeurIPS.
Candemir, S., et al., 2013. Lung segmentation in chest radiographs using anatomical atlases with nonrigid registration. IEEE TMI 33, 577–590.
Codella, N.C., et al., 2018. Skin lesion analysis toward melanoma detection. In: IEEE ISBI.
Cohen, J.P., Morrison, P., Dao, L., Roth, K., Duong, T.Q., Ghassemi, M., 2020. Covid-19 image data collection: Prospective predictions are the future. arXiv:2006.11988.
Combalia, M., et al., 2019. Bcn20000: Dermoscopic lesions in the wild. arXiv:1908.02288.
Dayan, I., et al., 2021. Federated learning for predicting clinical outcomes in patients with COVID-19. Nature Med. 27.
Delange, M., et al., 2021. A continual learning survey: Defying forgetting in classification tasks. IEEE PAMI.
Evans, D., et al., 2018. A pragmatic introduction to secure multi-party computation. Found. Trends Priv. Secur. 2, 70–246.
Feki, I., et al., 2021. Federated learning for COVID-19 screening from chest x-ray images. Appl. Soft Comput. 106, 107330.
Gao, Z., Wu, F., Gao, W., Zhuang, X., 2022. A new framework of swarm learning consolidating knowledge from multi-center non-iid data for medical image segmentation. IEEE TMI 1.
Geiping, J., et al., 2020. Inverting gradients-how easy is it to break privacy in federated learning? In: NeurIPS.
Goodfellow, I.J., et al., 2013. An empirical investigation of catastrophic forgetting in gradient-based neural networks. arXiv:1312.6211.
Goodfellow, I., et al., 2014. Generative adversarial nets. In: NeurIPS.
Heusel, M., et al., 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: NeurIPS.
Irvin, J., et al., 2019. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In: AAAI.
Jaeger, S., et al., 2013. Automatic tuberculosis screening using chest radiographs. IEEE TMI 33, 233–245.
Jaeger, S., et al., 2014. Two public chest x-ray datasets for computer-aided screening of pulmonary diseases. Quant. Imaging Med. Surg. 4, 475.
Kairouz, P., et al., 2021. Advances and open problems in federated learning. Found. Trends Mach. Learn. 14.
Karras, T., et al., 2020a. Analyzing and improving the image quality of stylegan. In: CVPR.
Karras, T., et al., 2020b. Training generative adversarial networks with limited data. In: NeurIPS.
Lalitha, A., et al., 2019. Peer-to-peer federated learning on graphs. arXiv:1901.11173.
Li, X., Jiang, M., Zhang, X., Kamp, M., Dou, Q., 2021. Fedbn: Federated learning on non-iid features via local batch normalization. arXiv:2102.07623.
Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V., 2020. Federated optimization in heterogeneous networks. In: Proceedings of Machine Learning and Systems, Vol. 2. pp. 429–450.
Li, W., et al., 2019. Privacy-preserving federated brain tumour segmentation. In: International Workshop on Machine Learning in Medical Imaging. Springer, pp. 133–141.
Lian, X., et al., 2017. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In: NeurIPS.
McMahan, B., et al., 2017. Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics. PMLR, pp. 1273–1282.
Mirza, M., Osindero, S., 2014. Conditional generative adversarial nets. arXiv:1411.1784.
Ratcliff, R., 1990. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. Psychol. Rev. 97, 285.
Robins, A., 1995. Catastrophic forgetting, rehearsal and pseudorehearsal. Connect. Sci. 7, 123–146.

Rolnick, D., et al., 2019. Experience replay for continual learning. In: NeurIPS.
Roy, A.G., et al., 2019. Braintorrent: A peer-to-peer environment for decentralized federated learning. arXiv:1905.06731.
Shoham, N., et al., 2019. Overcoming forgetting in federated learning on non-iid data. arXiv:1910.07796.
Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. In: ICLR.
Sun, T., Li, D., Wang, B., 2021. Decentralized federated averaging. arXiv:2104.11375.
Tschandl, P., Rosendahl, C., Kittler, H., 2018. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. Sci. Data 5, 1–9.
Wang, X., et al., 2017. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In: CVPR.
Wang, H., et al., 2020. Federated learning with matched averaging. arXiv:2002.06440.
Wink, T., Nochta, Z., 2021. An approach for peer-to-peer federated learning. In: 2021 51st Annual IEEE/IFIP DSN-W.
Xie, L., Lin, K., Wang, S., Wang, F., Zhou, J., 2018. Differentially private generative adversarial network. arXiv:1802.06739.
Yang, Q., et al., 2019. Federated machine learning: Concept and applications. ACM TIST 10, 1–19.
Zech, J.R., et al., 2018. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. PLoS Med. 15.
Zhang, R., et al., 2018. The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR.
Zhao, B., et al., 2020. IDLG: Improved deep leakage from gradients. arXiv:2001.02610.
Zhu, L., et al., 2019. Deep leakage from gradients. In: NeurIPS.