Contents lists available at ScienceDirect

Computers in Biology and Medicine

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

Personalized and privacy-preserving federated heterogeneous medical image analysis with PPPML-HMI

Juexiao Zhou [a,b,1], Longxi Zhou [a,b,1], Di Wang [a,b], Xiaopeng Xu [a,b], Haoyang Li [a,b], Yuetan Chu [a,b], Wenkai Han [a,b], Xin Gao [a,b,*]

[a] Computer Science Program, Computer, Electrical and Mathematical Sciences and Engineering Division, King Abdullah University of Science and Technology (KAUST), Thuwal, 23955-6900, Kingdom of Saudi Arabia

[b] Computational Bioscience Research Center, Computer, Electrical and Mathematical Sciences and Engineering Division, King Abdullah University of Science and Technology (KAUST), Thuwal, 23955-6900, Kingdom of Saudi Arabia

ARTICLE INFO

Keywords:
Medical imaging
Personalization
Federated learning
Privacy

ABSTRACT

Heterogeneous data is endemic due to the use of diverse models and settings of devices by hospitals in the field of medical imaging. However, there are few open-source frameworks for federated heterogeneous medical image analysis with personalization and privacy protection without the demand to modify the existing model structures or to share any private data. Here, we proposed PPPML-HMI, a novel open-source learning paradigm for personalized and privacy-preserving federated heterogeneous medical image analysis. To our best knowledge, personalization and privacy protection were discussed simultaneously for the first time under the federated scenario by integrating the PerFedAvg algorithm and designing the novel cyclic secure aggregation with the homomorphic encryption algorithm. To show the utility of PPPML-HMI, we applied it to a simulated classification task namely the classification of healthy people and patients from the RAD-ChestCT Dataset, and one real-world segmentation task namely the segmentation of lung infections from COVID-19 CT scans. Meanwhile, we applied the improved deep leakage from gradients to simulate adversarial attacks and showed the strong privacy-preserving capability of PPPML-HMI. By applying PPPML-HMI to both tasks with different neural networks, a varied number of users, and sample sizes, we demonstrated the strong generalizability of PPPML-HMI in privacy-preserving federated learning on heterogeneous medical images.

1. Introduction

Data-hungary artificial intelligence (AI), including various machine learning (ML) and deep learning (DL) methods [1], is increasingly being applied to solve miscellaneous problems in medical image analysis (MIA) and has led to disruptive innovations in pathology, radiology, and other fields [2–8]. Since modern DL models typically have millions of parameters or even more [9], a mass of curated data is usually required to train such data-hungry models to achieve clinical-grade performance [10–12]. However, even with modern advanced data science, generating a huge amount of data to fulfill the requirements of training models independently is still challenging for most hospitals and clinics. Therefore, seeking the cooperation of institutions to jointly generate data and train a joint model becomes an ideal solution [13]. In the centralized training, the server needs to collect data from all collaborators, and then the ML/DL model will be trained on the server. Nevertheless, such a strategy leads to more concerns related to data security and privacy. For example, training an AI-based lung infection detector [14–17] required a large amount of high-quality computerized tomography (CT) scans and human-labeled metadata, while in reality, such data is difficult to be obtained and shared because health data is usually highly sensitive thus usage is tightly regulated [5,18]. Hence, federated learning (FL) [19–22] was proposed as a learning paradigm that aims to address data governance and privacy issues by collaboratively training models without the need to share the data itself.

* Corresponding author. Computer Science Program, Computer, Electrical and Mathematical Sciences and Engineering Division, King Abdullah University of Science and Technology (KAUST), Thuwal, 23955-6900, Kingdom of Saudi Arabia.

E-mail address: xin.gao@kaust.edu.sa (X. Gao).

[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.compbiomed.2023.107861
Received 24 July 2023; Received in revised form 13 December 2023; Accepted 14 December 2023
Available online 19 December 2023

1.1. Federated learning

In FL, it is assumed that a set of n ($n \geq 2$) users are connected to a server, where each user can only access its own data [19]. Upon that, the users' goal is to acquire a model that captures the features of all users' data without sharing their local data with any other user or the server. Though each user can solely train the model with its own data without sharing any information with other users, the independently trained model of each user may not generalize well to other users' data or new samples, especially in the case of strong heterogeneity. Thus the following FL procedure was devised to learn a more generalized server model. Firstly, all users will receive a copy of the current server model and update the local model using its own data. After that, users send the updated model to the server. Finally, the server aggregates received local models to update the server model for the next broadcasting. This process continues until a generalized server model could be generated [20,23]. To be more specific, McMahan et al. [20] proposed the federated averaging (FedAvg) algorithm, which is the most famous aggregation method in the community, to aggregate local models collected by the server. Building upon this, recent studies also focused on crafting practical and robust federated learning frameworks designed for real-world applications, catering to various deep learning models across diverse platform architectures, such as Flower [24], EasyFL [25], FATE [26], R2Fed [27], and so on [28–30].

However, previous studies showed that the FedAvg algorithm might not converge or could be slowed down when local models drift significantly from each other due to the heterogeneity of local non-independent and identically distributed (non-IID) data [31]. Therefore, in the presence of heterogeneity, the server model trained by FL may not generalize well to each user's data [32], which is a significant obstacle to applying FL in practice. Taking the infection segmentation on CT scans as an example, different hospitals may have diverse CT scanners and scanning settings, thus the CT scans will have inherent diversity. With that, the server model trained with FL for segmentation will be unable to achieve good performance on each user's data due to heterogeneity.

1.2. Personalized federated learning

To apply the FL paradigm with the heterogeneous data as in the case of CT diagnosis, personalized federated learning (PFL) was devised as an enhanced version of FL [33–38]. To address personalization in FL, a two-step approach namely 'FL training + local adaptation' was regarded as the most commonly acknowledged strategy by the FL community [36, 39]. With this strategy, the server model is firstly trained using FL on heterogeneous users' CT data. Unexpectedly, the server model may perform poorly on each user's data due to data heterogeneity. Therefore, a few additional training steps are required to adapt this server model locally and realize the personalization. Depending on the specific strategies used in training, different personalized variants of the FedAvg algorithm were proposed, such as pFedMe [35], Per-FedAvg [34], and APFL [40]. However, all the aforementioned methods were theoretical research, and little applied attempt has been conducted, especially in the medical analysis field [41,42].

In addition to optimizing the training strategies for heterogeneous data, FedAVG + Share [43] improves performance on non-IID data by sharing a small amount of data among users. However, the strategy could not be adopted when the user's data is required to be strictly private. FedReplay [44] needs to train a universal and auxiliary encoder network, which encodes each user's data into latent variables that will be used to train the server model for classification. Therefore, given an existing neural network model, e.g. a segmentation model, it needs to be disassembled and restructured to work with FedReplay in FL. Thus, FedReplay cannot be simply and directly combined with the existing models and also might not be easily used by new users as a closed source method. As the latest work, FedPerGNN [45] was specially designed for graph neural networks and thus was limited to the application of graph data, which is usually different from medical imaging data. Therefore, with strictly prohibited raw data sharing and without any structural modification to the existing DL methods, an open-source, user-friendly, plug-and-play, and robust privacy-preserving framework for personalized federated heterogeneous medical imaging tasks is necessary.

1.3. Privacy in FL and PFL

Privacy is a hot and significant topic in the age of medical big data [46]. Nevertheless, when we applaud the fact that private medical data is no longer shared with other parties in FL, previous studies showed that FL is still vulnerable to attacks, such as data poisoning attack [47], membership inference attack [48–50], source inference attack (SIA) [51], attribute reconstruction attack [52], and inversion attack [53–56], thus compromising data privacy.

Due to privacy concerns associated with the presence of a server in traditional FL, especially when the client does not trust the server, the concept of decentralized federated learning (DFL) has been introduced. DFL operates as a decentralized structure, wherein clients communicate and exchange model parameters directly without relying on a central server [57,58]. Various DFL methods have been proposed, such as peer-to-peer FL [59], server free FL [60], serverless FL [61], device-to-device FL [62], and swarm learning [63]. However, DFL is still facing a series of challenges such as high communication overhead, network security, and all [64] compared to conventional FL. Meanwhile, the majority of existing DFL approaches primarily contribute to theoretical research, with a limited focus on applications in medical imaging and addressing data heterogeneity.

Similar to FL, PFL is also facing the threat of privacy attacks. Diverse strategies could be used to protect data privacy with FL and PFL. As the latest research, differential privacy (DP) [65] was used to add noise to the gradient transmitted in PFL [45,66] as same as in FL [67] to protect the privacy of users' data. However, DP adopts the mechanism of adding noise to enhance privacy protection while sacrificing model accuracy [68], resulting in difficulty to achieve clinical-grade performance in practice. Besides, cryptographic approaches, including secure aggregation (SA) [69], homomorphic encryption (HE) [70], multi-party computation (MPC) [71] and etc., realize privacy preservation by sacrificing time and space without affecting the accuracy much. Among these techniques, MPC requires the involvement of multiple servers, which is different from the case with only one server. Conventional SA requires heavy communications between users and the server, which will cause overhead for users with limited resources, while the latest decentralized version of SA [72] transferred the process of aggregation from the server to users, reduced the need for high communication between all users and the server, and further strengthened the privacy protection as the server is usually an un-trusted third party. However, the protocol used in Ref. [72] required a three-step process as 'decryption-summation-encryption' to aggregate the local gradients into the transmitted gradients in the loop, which could be further simplified into a two-step process as 'encryption-summation' with the help of HE as in PPPML-HMI.

To strengthen privacy protection in FL, several open-source methods, such as FATE [26], PySyft [73], and NVFlare [74], have already been developed to secure gradients during training using techniques like SA and HE. However, these methods are mainly based on the conventional FL and thus could not address the challenges with heterogeneous data while still allowing privacy protection.

1.4. Heterogeneity and privacy in MIA

As a representative of many medical tasks that would strongly benefit from personalization and privacy protection, especially in medical imaging, the accurate detection and segmentation of lung infection caused by the severe acute respiratory syndrome coronavirus 2

(SARS-CoV-2, COVID-19) have been such an important task since 2020 [75]. To diagnose lung diseases, imaging is the major source of data and the most commonly used imaging technologies are X-rays and CT scans [76]. Meanwhile, DL has been widely applied in developing the computer-aided diagnosis (CAD) systems for COVID-19 [8,15,77–81]. Most aforementioned works require centralized training, where the research institution acts as a coordinator/server to collect raw CT scans from users like hospitals to train a model centrally. Several problems exist in this process. Firstly, all users need to strictly trust each other and the server in order to share the raw CT scans, which might limit the number of users and available data involved, leading to insufficient training data. Secondly, users have to deliver the private raw CT scans to the server for centralized training, leading to potential privacy breaches. Therefore, a proper FL approach is necessary to allow users to keep their data private, thus more data providers could participate in the co-training of the model and more diverse data could be used to train a

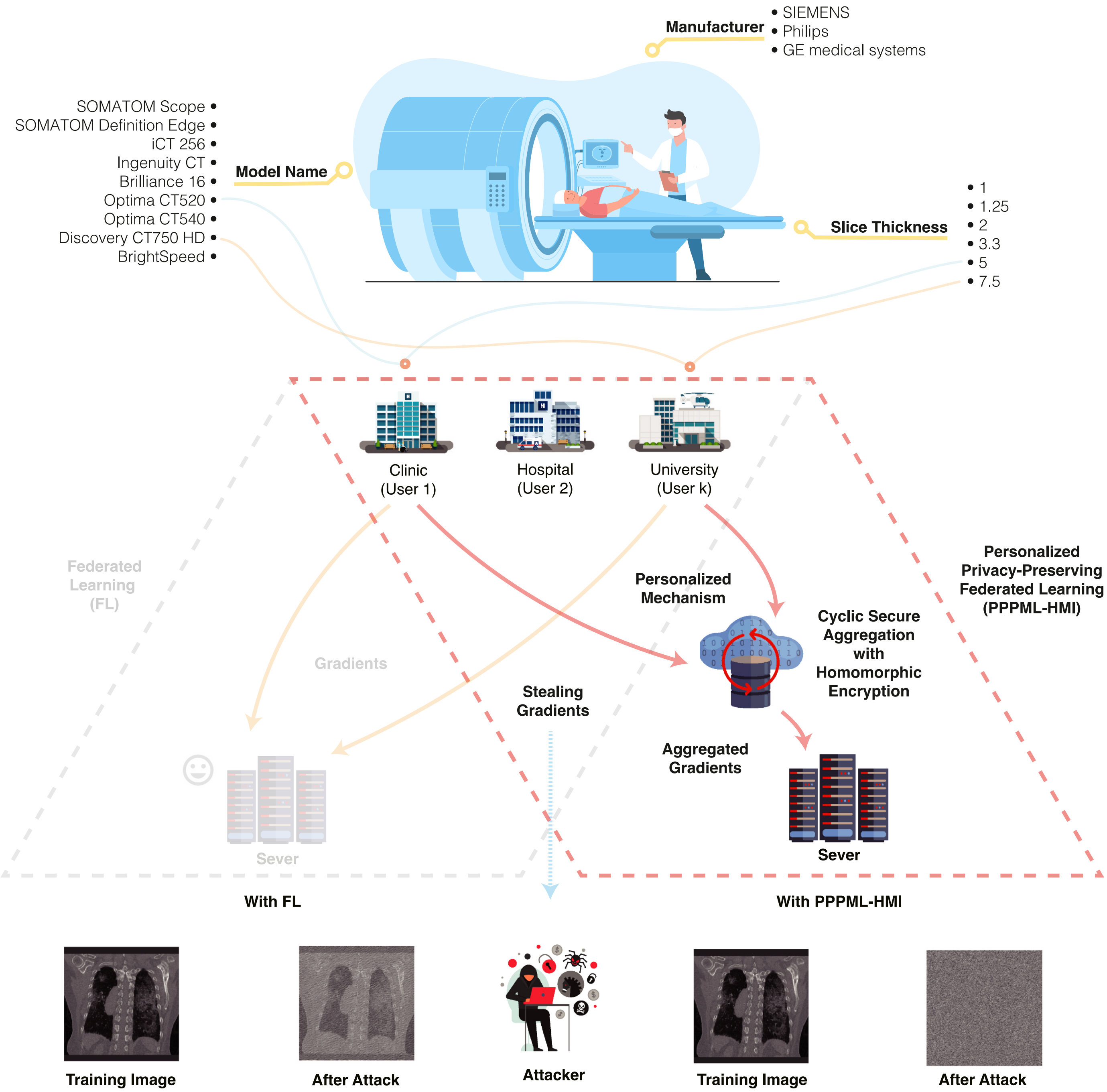


Fig. 1. Scheme of PPPML-HMI. In our real-world case, hospitals (user 1 to user k) use devices from different manufacturers, models, and settings for the detection of lung infection by COVID-19. The use of diverse devices generates data with inherent differences, namely heterogeneous data. With FL (indicated by grey dashed lines), the goal is to jointly train a consensus model with the data from each hospital without sharing the data itself. With homogeneous data across hospitals, FL could efficiently train a server model that works well for all hospitals. However, when hospitals have heterogeneous data, the server model trained by FL could not perform well when applied to each hospital. Thus, a personalized mechanism was integrated into PPPML-HMI and allowed models to adapt to heterogeneous data (indicated by pink dashed lines). To strengthen the privacy protection of PPPML-HMI, we designed the cyclic secure aggregation with homomorphic encryption to transfer the process of aggregating gradients from the server to the users in a decentralized manner. To demonstrate the privacy-preserving capability of PPPML-HMI, we simulated that an attacker could steal the gradients passed in FL and try to recover the original picture.

model with better generalization power. Though FL has been widely applied in tasks related to COVID-19 in the latest research [82–87], the heterogeneous data and privacy breaches are still problems as there is currently no such open-source solution for personalized and privacy-preserving federated heterogeneous medical image analysis, especially for the heterogeneous COVID-19 CT analysis.

Here, we proposed PPPML-HMI, a novel open-source, robust, user-friendly and plug-and-play method for personalized and privacy-preserving federated heterogeneous medical image analysis (Fig. 1). PPPML-HMI specifically targets the scenario where no raw data should be shared with any third party, no structural modification should be conducted to existing DL models, and in a context of heterogeneous data. To our best knowledge, personalization and privacy protection were discussed simultaneously for the first time under the federated scenario by integrating the PerFedAvg algorithm and designing the novel cyclic secure aggregation algorithm with homomorphic encryption (CSAHE). To demonstrate the utility of PPPML-HMI, we applied it to a simulated classification task namely the classification of healthy people and patients from the RAD-ChestCT Dataset [88,89], and one real-world segmentation task namely the segmentation of lung infections from COVID-19 CT scans by extending our previous method for the task [8,15], which was also a general method for segmentation of lung, tracheal, vascular, etc. By applying PPPML-HMI to both tasks with different neural networks, a varied number of users, and sample sizes, we further demonstrated the strong generalizability of PPPML-HMI. Finally, we also applied the improved deep leakage from gradients to simulate adversarial attacks and showed the strong privacy-preserving capability of PPPML-HMI.

2. Methods

2.1. Design of PPPML-HMI

PPML-HMI is built up with two major modules as a training framework: the PFL (Section 2.4) and the CSAHE modules (Fig. 1, Algorithm 1 and Section 2.5). During each round of the global training, the server broadcasts the server model to each user for initialization. Then, each user trains the local model using the local private data. After finishing the local training, each user calculates the gradient between the local model and the server model. To avoid sending users' gradients directly to the server, which could lead to potential privacy leakage, PPPML-HMI transfers the gradient aggregation process that is originally performed on the server to a loop composed of all users through the CSAHE mechanism in a decentralized manner. At the end of each global training, the CSAHE mechanism is executed. A user in the loop will be randomly selected as the initiator, who will protect its own gradient by summing a random mask as noise to the gradient and encrypting the noised gradient with HE, and will transmit the noised gradient into the loop for further aggregation. The noise in the aggregated gradients is kept till the end of the execution of the CSAHE mechanism. PPPML-HMI achieves decentralized secure gradient aggregation with homomorphic encryption (Section 2.5), thus each user could confidently aggregate their own gradients to the transmitted gradient without worrying about privacy issues. The code for this paper is publicly available at https://github.com/JoshuaChou2018/PPPML-HMI.

2.2. Dataset processing

For the classification task, we simulated and constructed our heterogeneous data from the RAD-ChestCT Dataset [88,89], which includes 35,747 chest CT scans from 19,661 adult patients. For the segmentation task, we collected 180 anonymized CT scans generated by diverse CT scanners and scanning parameters from five hospitals labeled as A ∼ E. All patients were confirmed to be COVID-19 positive by either the nucleic acid test or antibody test.

To perform the segmentation of lung infections from the 3D CT scans, we need to find a mapping $F : R^{(H\times W\times S)} \mapsto \{0,1\}^{H\times W\times S}$, where $H \times W$ is the height and width of each 2D CT image and S is the number of images. Since the data generated by different CT scanners owned various volume sizes, spatial normalization was adopted to re-scale raw CT data into a machine-agnostic standard space with a fixed shape ($512 \times 512 \times 512$). As in Ref. [15], we decomposed the 3D segmentation of each 3D CT scan into three 2D segmentation problems along the x-y, y-z, and x-z views (axial, sagittal, and coronal). The training along each plane was performed independently.

All prediction and visualization of samples were performed with models trained with data excluding the corresponding sample itself.

2.3. Neural network for the segmentation of lung infections

For the classification task, we adopted the 3D DenseNet [90] as the backbone DL model. For each segmentation task along the three views, we trained an independent U-Net that took five adjacent images with dimension: $R^{5\times512\times512}$ as inputs, and output the probability map of infection regions for the central image with dimension: $R^{512\times512}$. The U-Net for 2D segmentation consisted of four encoding layers, one bottleneck layer, and four decoding layers as shown in Fig. 2A.

Given a 3D CT scan, we applied three 2D U-Net models and generated three segmentation results p_{xy}, p_{yz}, p_{xz} along the x-y, y-z, and x-z views. The final segmentation result in 3D space was calculated by summing up three intermediate predictions followed by taking a threshold of 2 as $p_{final} = (p_{xy} + p_{yz} + p_{xz}) \geq 2$.

2.4. Personalized federated learning

To accomplish personalized FL, the personalized FedAvg (Per-FedAvg) [34] algorithm was adopted to acquire the optimal initial model (meta-model) as the server model, which could be easily adapted to the local heterogeneous data by performing just a few steps of gradient descent. Per-FedAvg was inspired by the fundamental idea of the Model-Agnostic Meta-Learning (MAML) framework [91]. Given a set of tasks from different underlying distributions, instead of finding the model that generalizes on all tasks as FL, MAML tends to find a meta-model that could perform better in different tasks after a few steps of local gradient descent.

In FL, the goal of optimization is:

$$\min_{w\in R^d} f(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w)$$

where $f_i(w)$ is the loss function to user u_i.

With the concept of MAML, the goal of the optimization becomes finding a good initialization:

$$\min_{w\in R^d} F(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w - \alpha\nabla f_i(w))$$

where α (α ≥ 0) is the step size.

As shown in Algorithm1, at each epoch k, the server will broadcast the server model to all users. Then, all users will train their local model with τ local epochs. After τ local epochs, a list of $\{w^i_{k+1,t}\}^{\tau}_{t=0}$ will be generated with respect to the user u_i, where $w^i_{k+1,0} = w_k$, $w^i_{k+1,t} = w^i_{k+1,t-1} - \beta\widetilde{\nabla}F_i(w^i_{k+1,t-1})$, β is the local learning rate and $\widetilde{\nabla}F_i(w^i_{k+1,t-1})$ is an estimate of $\nabla F_i(w^i_{k+1,t-1})$.

Algorithm 1. PPPML-HMI

2.5. Cyclic secure aggregation with homomorphic encryption

Secure aggregation (SA) was introduced for FL by Bonawitz et al. [69], in which they used blinding with random values namely Shamir's

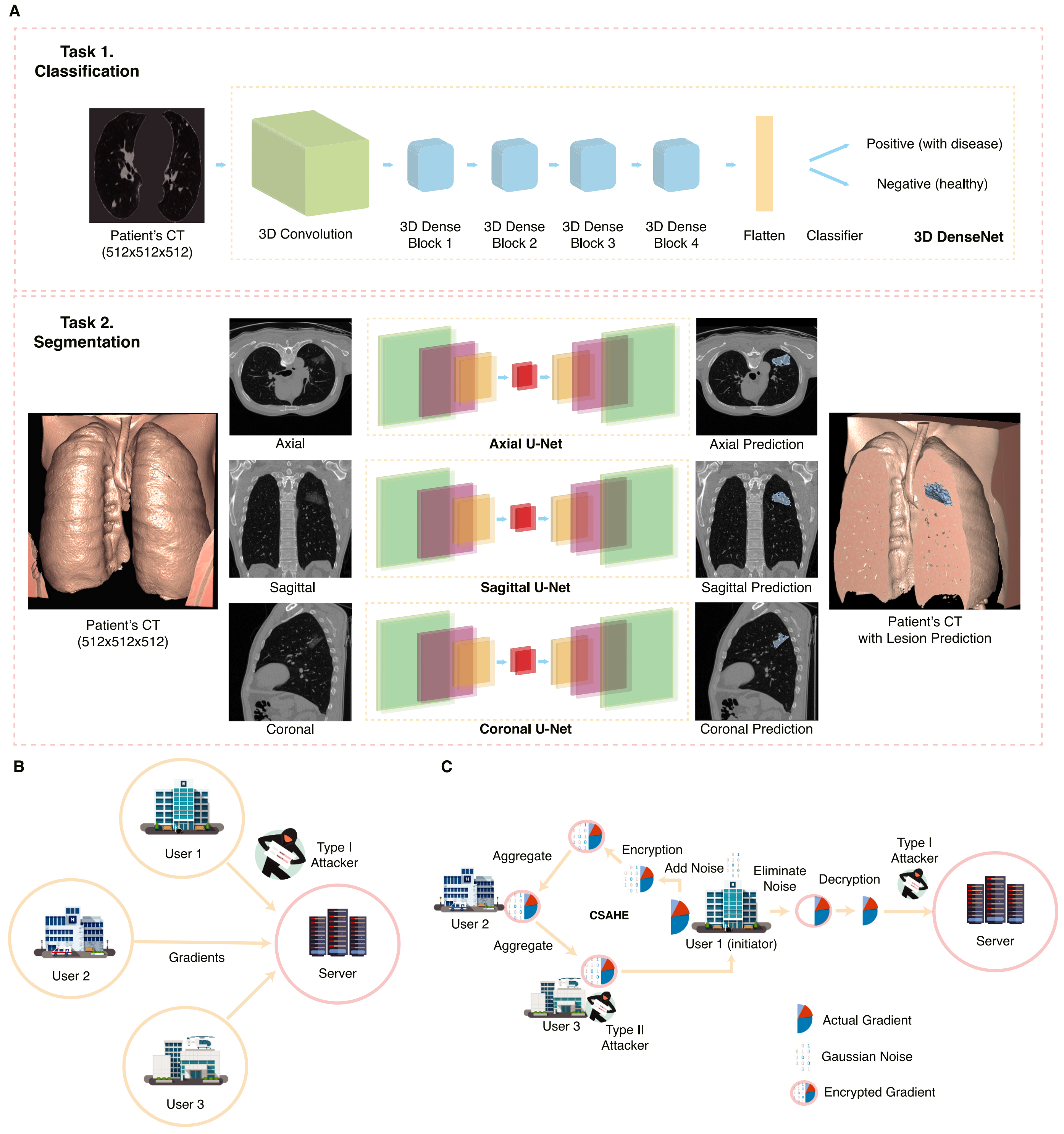


Fig. 2. A) Illustration of two tasks. We applied PPPML-HMI to the classification of healthy people and patients with 3D DenseNet on the RAD-ChestCT Dataset, and the segmentation of the lung infections of COVID-19 with a 2.5D U-Net method [8,15]. Illustration of the communication network and attackers of FL (B) and PPPML-HMI (C). Two types of attackers exist in our setting: 1) Attackers who can intercept messages sent from any users to the server or between users (type I), and 2) Honest-but-curious attackers who are part of the users of PPPML-HMI (type II).

Secret Sharing (SSS) [92] and symmetric encryption to protect the local models. The fundamental idea of SA is simple: suppose we have two users holding two private numbers a and b, and the server needs to calculate the value of $a+b$ without knowing the actual value of a and b. Then we could generate a random number r and calculate $a^{'} = a + r$ and $b^{'} = b - r$. Hence the server could get the sum of a and b without knowing the actual value of a and b respectively. Based on this idea, Bonawitz et al. [69] let any two users u_i and u_j in the FL share a random number r_{ij} to mask the actual model weight with $w_i^{'} = mask(w_i, r_{ij})$ and $w_j^{'} = mask(w_j, r_{ij})$, thus the server can only get the $w_i^{'}$ and $w_j^{'}$ from users without knowing the real values of w_i and w_j. Still, the server can get the accurate summation when performing the aggregation as $w_i^{'} + w_j^{'} = w_i + w_j$. However, their protocol requires four communications between each

```
Algorithm 1 PPPML-HMI
Require: $K$ is the number of global epochs; $\tau$ is the number of local epochs; $N$ is the number of users; $\gamma$ is the number of final adaptation epochs; $w_k$ is the parameters of the server model at global epoch $k$; $w_{k,t}^i$ is the parameters of the model of user $u_i$ at local epoch $t$ and global epoch $k$; $\gamma$ is the number of epochs for the final adaptation; $d_i$ is the number samples of user $u_i$; $D$ is the total number of samples of all users.
Procedure: System Starts
For $k = 0, \dots, K-1$ (Global epoch)
    Server sends $w_k$ to all users
    For each user $u_i$ with $i = 0, .., N-1$
        Set $w_{k+1,0}^i = w_k$
        For $t = 0, \dots, \tau - 1$ (Local epoch)
            Compute the stochastic gradient $\tilde{\nabla} f_i\left(w_{k+1,t-1}^i, D^i\right)$ using dataset $D^i$
            Set $\tilde{w}_{k+1,t}^i = w_{k+1,t-1}^i - \alpha\tilde{\nabla} f_i\left(w_{k+1,t-1}^i, D^i\right)$
            Set $w_{k+1,t}^i = w_{k+1,t-1}^i - \beta\left(I - \alpha\tilde{\nabla} f_i\left(w_{k+1,t-1}^i, D''^i\right)\right)\tilde{\nabla} f_i\left(\tilde{w}_{k+1,t}^i, D'^i\right)$
        Each user $u_i$ with $i = 0, .., N-1$ calculates the weighted gradient compared to the server $\Delta w_{k+1,\tau}^i = \left(w_{k+1,\tau}^i - w_k\right) * \frac{d_i}{D}$
        Execute cyclic secure aggregation with $\boldsymbol{CSAHE}(u_0, \dots, u_{N-1})$
        Initiator $u_I$ sends securely aggregated update $\Delta w_{k+1,\tau}^{CSA}$ back to the server
    Server updates its model with the received gradients $w_{k+1} = w_k + \Delta w_{k+1,\tau}^{CSA}$
Server broadcasts the meta-model to all users
For each user $u_i$ with $i = 0, .., N-1$
    For $t = 0, \dots, \gamma - 1$ (Final local adaptation)
        User $u_i$ adapts the meta-model with local private data

function CSAHE($u_0, \dots, u_{N-1}$)
Organize all users in a loop. Randomly select a user $u_I$ with $I \in \{0, \dots, N-1\}$ as the initiator, generate a random mask $\boldsymbol{R}$ with the same shape as the gradient $\Delta w_{k+1,\tau}^I$ hold by $u_I$ from Gaussian Noise with a large $\sigma$.
Suppose user $u_r$ is the initiator. User $u_r$ adds the $\boldsymbol{R}$ to the gradient as $\Delta w_{k+1,\tau}^{CSA} = \Delta w_{k+1,\tau}^I + \boldsymbol{R}$ and encrypt it with homomorphic encryption (HE) as $HE\left(\Delta w_{k+1,\tau}^{CSA}\right)$
For each user $u_j$ in the circular chain, until $j = I$
    $HE\left(\Delta w_{k+1,\tau}^{CSA}\right) = HE\left(\Delta w_{k+1,\tau}^{CSA}\right) + HE\left(\Delta w_{k+1,\tau}^j\right)$
    User $u_j$ transfers the $HE\left(\Delta w_{k+1,\tau}^{CSA}\right)$ to the next user in the loop
The initiator $u_I$ removes the random mask from the final aggregated gradient with $HE\left(\Delta w_{k+1,\tau}^{CSA}\right) = HE\left(\Delta w_{k+1,\tau}^{CSA}\right) - HE(\boldsymbol{R})$ and decrypts the encrypted aggregated gradient $HE\left(\Delta w_{k+1,\tau}^{CSA}\right)$ into $\Delta w_{k+1,\tau}^{CSA}$.
Return the securely aggregated gradient $\Delta w_{k+1,\tau}^{CSA}$
```

user and the server, which will cause overhead for users with limited resources.

Based on the aforementioned possibilities for improvement, we designed the cyclic secure aggregation with homomorphic encryption (CSAHE) algorithm to transfer the secure aggregation from the server to a loop composed of all users in a decentralized manner with a two-step process as 'encryption-summation' for all non-initiator users. To protect the gradients transmitted in CSAHE, we encrypted all gradients with homomorphic encryption (MHE) based on the Cheon-Kim-Kim-Song (CKKS) cryptographic scheme [93] that provides approximate arithmetic over vectors of complex numbers and performed the aggregation homomorphically with TenSEAL [94], which is a python library for performing homomorphic encryption operations on tensors, built on top of Microsoft SEAL.

We integrated the CSAHE into PPPML-HMI to protect the gradients of users. As shown in Algorithm1, in each epoch, all users form a loop and an initiator is selected randomly from all users, while the remaining users are called non-initiator users. The initiator generates a random mask using a Gaussian distribution with a self-defined large σ. After that, the random mask will be summed to the initiator's gradient to protect its actual value. Then, the noised gradient will be homomorphically encrypted and transmitted to the next user in the loop, who can also safely aggregate its gradient to this transmitted gradient homomorphically, and transmit the newly aggregated gradient to the next user. The aforementioned process keeps working till the aggregated gradient is transmitted back to the initiator. Then, the initiator eliminates the random mask from the aggregated gradient and decrypts it to recover the actual value. Finally, the initiator sends the aggregated gradient to the server for updating the server model.

With CSAHE, each user except the initiator only needs to interact with two users located before and after in the loop and does not need to have other interactions with the server except for downloading the updated model in each global epoch. The secure aggregation in PPPML-HMI is conducted in a decentralized manner, which is different from the conventional SA that happens at the server. HE in CSAHE allows all non-initiator users to aggregate their gradients homomorphically without the need to decrypt. As shown in Algorithm1, once the $w_{k+1,\tau}^i$ of each user u_i is calculated, user u_i will calculate the weighted difference between the $w_{k+1,\tau}^i$ and the server model w_k as $\triangle w_{k+1,\tau}^i = \left(w_{k+1,\tau}^i - w_k\right) * \frac{d_i}{D}$. Then, all $\triangle w_{k+1,\tau}^i$ with $i = 0, \dots, N-1$ will be securely aggregated through the CSAHE algorithm. Finally, the server will collect the securely aggregated gradient $\Delta w_{k+1,\tau}^{CSA}$ from the initiator for updating the server model with $w_{k+1} = w_k + \Delta w_{k+1,\tau}^{CSA}$.

The public key encryption scheme is used in CSAHE, where a public key and a secret key are generated (Fig. 2C). The public key is shared by all users to allow the encryption of gradients before the homomorphic aggregation and the private key is held only by the initiator to allow the decryption of the aggregated gradient before sending it to the server. Since the initiator is randomly selected in each epoch, the public key and secret key will be re-generated before the homomorphic aggregation during each epoch. While it is true that under normal circumstances, a non-initiator does not have access to the private key, it's essential to address the potential vulnerabilities of CSAHE in practical scenarios where the private key could be compromised. In such unfortunate situations, a non-initiator may gain unauthorized access to the private key, thereby enabling them to decrypt the homomorphically encrypted gradient in the event of an attack. This vulnerability underscores the importance of safeguarding cryptographic keys, especially the private key held by the initiator, to reduce the risk of privacy breaches. Still, we could agree that the mechanism of regenerating key pairs at each epoch in CSAHE could effectively protect the process of gradient aggregation and reduce the risk of private key leakage.

2.6. *iDLG reconstruction attack*

Previous research has shown that the gradients transmitted from the user to the server in FL may still compromise data privacy [53–55]. Among those studies, the improved deep leakage from gradient (iDLG) is a state-of-the-art approach to obtain private training data from the gradients transmitted between users and the server as shown in Algorithm2.

Algorithm 2. iDLG

2.7. *Performance evaluation*

To evaluate the performance on the classification task, we used the accuracy as defined below because the number of two classes is balanced:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where TP, TN, FP, and FN stand for true positive, true negative, false positive and false negative.

To evaluate the performance of lung infection segmentation with different methods, we used the Dice score as defined below:

$$Dice = \frac{2|Y \cap Y^{'}|}{|Y| + |Y^{'}|}$$

where Y is the actual infection region annotated by radiologists, $Y^{'}$ is the predicted infection region, and |Y| represents the cardinality of Y.

2.8. *Hyper-parameter selection and training settings*

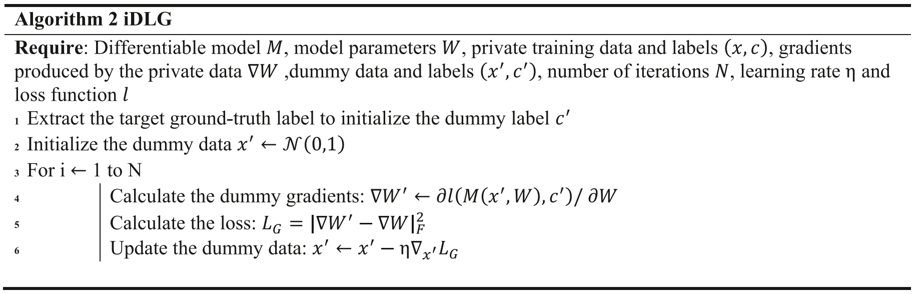
To ensure a fair comparison, we tested the number of global epochs from {10, 20, 50, 100} and the number of local epochs from {1, 5, 10, 20}. From our pre-experiments, we noticed that the number of global epochs $K = 20$ and the number of local epochs $\tau = 10$ enabled the model to converge and provided a good trade-off between the computation time and model performance. The batch size and learning rate were set to be 64 and 10^{-4} respectively. For CSAHE, the random noise was generated from a Gaussian distribution with mean = 0 and a randomly chosen large standard deviation (>100) to avoid the potential inversion attack. To implement HE with TenSEAL, we used the CKKS scheme with the polynomial modulus degree to be 8192 and the coefficient modulus sizes to be [26,26,26,26,26,26,31,31], meaning that the coefficient modulus will contain 8 primes of 31 bits, 26 bits, 26 bits, 26 bits, 26 bits, 26 bits, 26 bits, and 31 bits. During the training, 32 workers were used for data loading and processing on one machine with 120 GB RAM and one NVIDIA V100 GPU. Nested cross-validation was adopted for data splitting and model training. All experimental results presented are the average values of 5 experiments with random initialization.

3. Results

We applied PPPML-HMI to a simulated heterogeneous dataset from the RAD-ChestCT dataset, where users' data were grouped according to the slice thickness from 2 mm, 5 mm, 10 mm, to train a classification model and show the generalizability of PPPML-HMI when varying the number of users and the sample sizes. We also applied PPPML-HMI to a real-world case, where heterogeneous data were generated by five hospitals with different CT scanners, to train a segmentation model for COVID-19 lung infections. To demonstrate the effectiveness of PPPML-HMI, we compared three methods, including training independently using only each user's own data, training in a centralized training manner using complete data, and training with one of the most classical and famous algorithms in FL namely FedAvg [20]. With both tasks together with different neural networks, the number of users, and sample sizes, we further gave evidence of the robustness and generalizability of PPPML-HMI to these parameters. Meanwhile, we applied the improved deep leakage from gradients to simulate adversarial attacks on the segmentation task and showed the strong privacy-preserving capability of PPPML-HMI.

3.1. *PPPML-HMI is generalizable with various numbers of users and samples with varying data distribution in the classification task*

To show the generalizability of PPPML-HMI when varying the number of users, the sample sizes, and the data distribution, we simulated three sets of data partitions on the RAD-ChestCT dataset [88,89] according to the slice thickness of CT scans and labeled them as Split 1, Split 2, and Split 3 as shown in Table 1. Split 1 had only one user with 392 CT scans, which represented the centralized training scenario. Split 2 had two users with an equal number of CT scans (196 and 196) but with different slice thicknesses (2 mm and 5 mm). Split 3 had three users with a varied number of CT scans (174, 68, and 150) and slice thicknesses (2 mm, 5 mm, and 10 mm) simultaneously. Different slice thicknesses as a common heterogeneity of data affect the performance of model training in practice. For each split, the goal was to train models for classifying healthy people and patients based on CT scans. Compared



```
Algorithm 2 iDLG
Require: Differentiable model $M$, model parameters $W$, private training data and labels $(x, c)$, gradients produced by the private data $\nabla W$ ,dummy data and labels $(x', c')$, number of iterations $N$, learning rate η and loss function $l$
Extract the target ground-truth label to initialize the dummy label $c'$
Initialize the dummy data $x' \leftarrow \mathcal{N}(0,1)$
For i ← 1 to N
    Calculate the dummy gradients: $\nabla W' \leftarrow \partial l(M(x', W), c') / \partial W$
    Calculate the loss: $L_G = \|\nabla W' - \nabla W\|_F^2$
    Update the dummy data: $x' \leftarrow x' - \eta \nabla_{x'} L_G$
```

Table 1
Description of the RAD-ChestCT dataset and averaged predicted accuracy of methods.

Split ID	User ID	Slice thickness (mm)	#Patients	#Healthy People	Accuracy			
					Centralized training	Federated learning	PPPML-HMI (w/o CSAHE)	PPPML-HMI
Split 1	A	2	392	392	0.97	/	/	/
Split 2	A	2	196	196	0.95	0.68	0.94	0.94
	B	5	196	196				
Split 3	A	2	174	174	0.92	0.77	0.89	0.89
	B	5	68	68				
	C	10	150	150				

to centralized training, FL led to a drastic reduction in the averaged accuracy at Split 2 ($\triangle_{acc} = -0.27$) and Split 3 ($\triangle_{acc} = -0.15$), where PPPML-HMI showed less reduction and better performance compared to FL at Split 2 ($\triangle_{acc} = -0.01$) and Split 3 ($\triangle_{acc} = -0.03$). The ablation study (PPPML-HMI w/o CSAHE, which is the same as PerFedAvg) showed that the improved classification accuracy of PPPML-HMI was contributed by using PerFedAvg while CSAHE achieved privacy protection without compromising model effectiveness. These results provided evidence of the generalizability of PPPML-HMI when varying the number of users, the sample sizes, and the data distribution.

3.2. PPPML-HMI achieved personalization for federated heterogeneous segmentation of lung infections by COVID-19

Since the classification task was relatively easy, we further applied PPPML-HMI to a real-world case, namely the segmentation of lung infections by COVID-19. There were five hospitals in the real-world case, labeled as A ~ E, each of which used CT scanners of different models (Brilliance 16, iCT 256, Ingenuity CT, BrightSpeed, Optima CT520, Optima CT540, Discovery CT750 HD, SOMATOM Definition Edge, and SOMATOM Scope) from three manufacturers (Philips, GE medical systems, and SIEMENS). Hospitals also used different settings, such as slice thickness (1.00 nm, 1.25 nm, 2.00 nm, 3.30 nm, 5.00 nm, and 7.50 nm), and provided various numbers of data ranging from 9 to 119 as shown in Table 2. Different CT scanners and settings used by five hospitals led to inherent differences in the generated CT scans. As shown in Fig. 3A, we performed the dimension reduction and clustering on the original CT scan data provided by the hospital with the Principal Component Analysis (PCA) and assigned different colors to visualize the inherent differences between data according to the manufacturer, the hospital, and the slice thickness. Based on the results, CT scans generated by the CT scanners manufactured by GE medical system from hospital D showed significant differences from that of other manufacturers, including Philips and SIEMENS, and indicated that inherent differences existed in CT scans generated by different hospitals with diverse CT scanners.

To show the effectiveness of PPPML-HMI on federated heterogeneous medical image analysis, we compared different approaches, including the centralized training using complete data from all hospitals, the independent training using data from each hospital respectively, and the most classical and famous FL algorithm namely FedAvg as shown in Fig. 3. Without taking the data privacy issues into consideration, the centralized training worked best when we collected complete data from all users A ~ E to train the server model and performed the prediction on each user's data ($A_{Dice} = 0.64, B_{Dice} = 0.62, C_{Dice} = 0.51, D_{Dice} = 0.51, E_{Dice} = 0.69$) as shown in Fig. 3B. However, when the privacy issue matters, the server could not collect data from users, thus centralized training could not be performed. Moreover, since data exists in a distributed manner, the heterogeneous data might cause problems in the FL scenario. To train models with heterogeneous data in FL, as one solution, using each user's own data to train independent models resulted in a significant performance reduction ($\triangle_{Dice} = -0.08, -0.01, -0.07, -0.05, -0.03$ for users A ~ E respectively). Meanwhile, transferring a model trained on one user's data to another user showed even worse performance and indicated poor generalization ability of models trained with such a method, e.g. applying the model trained on user C to user D only yielded a Dice score of 0.28. These results provided further evidence of the strong heterogeneity in data across users in the real-world case.

Since training a model with each user's own data did not meet the need for clinical-grade performance, training a model using the information in all users' data but without sharing the raw data was necessary. With that, FL was the most intuitive solution. However, because of the strong heterogeneity of users' data, the server model trained by FL ($A_{Dice} = 0.51, D_{Dice} = 0.39, E_{Dice} = 0.63$) performed even worse on users A, D, and E than the independently trained model using only the users' own data ($A_{Dice} = 0.56, D_{Dice} = 0.46, E_{Dice} = 0.66$) as shown in Fig. 3C. Meanwhile, user C had only 9 samples and the data heterogeneity was not as significant as those between other users' data (Fig. 3A), thus the server model trained with FL could achieve similar performance as the centralized training with complete data only at user C (Fig. 3B and C).

In contrast to FL, similar to MAML, the server model generated by

Table 2
Description of the COVID-19 dataset. There are 5 hospitals (Ã E) and each of them owns data generated by different CT scanners and settings.

User ID	System label	Manufacturer	Model	Slice thickness (mm)	#Patients	Total
A	P_B16_2.0	Philips	Brilliance 16	2.00	5	12
	P_B16_3.3	Philips	Brilliance 16	3.30	1	
	P_B16_7.5	Philips	Brilliance 16	7.50	6	
B	P_I256_1.0	Philips	iCT 256	1.00	114	119
	P_I256_5.0	Philips	iCT 256	5.00	5	
C	P_I_1.0	Philips	Ingenuity CT	1.00	9	9
D	G_B_1.25	GE medical systems	BrightSpeed	1.25	1	24
	G_B_5.0	GE medical systems	BrightSpeed	5.00	4	
	G_CT520_1.25	GE medical systems	Optima CT520	1.25	11	
	G_CT540_1.25	GE medical systems	Optima CT540	1.25	7	
	G_CT750_5.0	GE medical systems	Discovery CT750 HD	5.00	1	
E	S_SDE_1.0	SIEMENS	SOMATOM Definition Edge	1.00	10	16
	S_SS_2.0	SIEMENS	SOMATOM Scope	2.00	6	

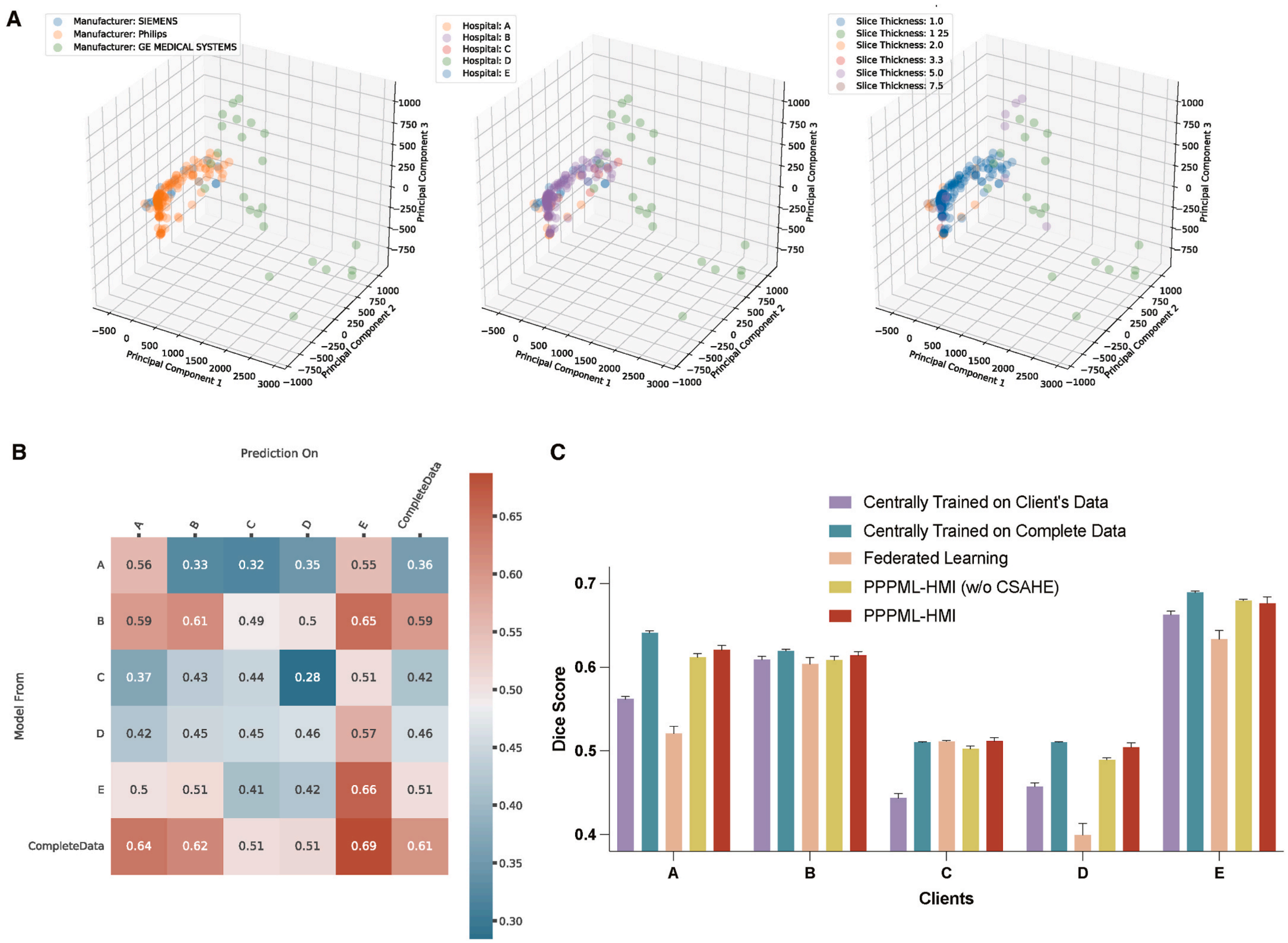


Fig. 3. A) Dimension reduction and clustering with PCA according to the manufacturer, originating hospital, and slice thickness indicated that CT scans generated by different CT scanners had significant inherent differences. B) Heatmap showed the Dice score of segmentation when applying models trained centrally on the data of each hospital. C) Barplot showed the Dice score of models trained centrally, with federated learning, and with PPPML-HMI.

PPPML-HMI was a good initialization, which could learn the common features in heterogeneous data and could be easily adapted to local user's data by a few local training steps. As shown in Fig. 3C, with the server model generated by PPPML-HMI as an initialization, only γ ($\gamma < 5$) steps of local training on the user's data could adapt the server model to local user's data ($A_{Dice} = 0.62$,$B_{Dice} = 0.61$,$C_{Dice} = 0.51$,$D_{Dice} = 0.51$, $E_{Dice} = 0.68$) and achieve a similar performance as the centralized training with complete data and better performance than FL under the same total number of epochs. Additionally, the improvement of PPPML-HMI compared to FL was most significant for users A and D, as the data of both users showed the most significant data heterogeneity from the data of other users as shown in Fig. 3A.

To further understand the differences in performance between methods, we visualized the predicted segmentation of the lung infections on a high-quality sample (A000069) and a low-quality sample (A000075), respectively. The low-quality sample had significantly worse clarity and resolution of CT images than the high-quality sample as shown in Fig. 4A. Orange, green, and yellow were used to represent true positives, false positives, and false negatives respectively compared to the ground truth. For A000069, the independently trained model on the associated user's data gave enormous false positives, while the model trained with FL predicted a large number of false negatives. In contrast, the personalized model from PPPML-HMI provided competitive performance as the centrally trained model with complete data. For A000075, the personalized model from PPPML-HMI predicted more true positives compared to the centrally trained model with the complete data and rescued more false negatives compared to the model trained with FL. In summary, PPPML-HMI allowed model personalization to each user with heterogeneous data and achieved competitive performance as the centralized training with complete data.

3.3. PPPML-HMI protects privacy of CT scans

There are two types of attackers in our setting: 1) attackers who can intercept messages sent from any users to the server and between any users (type I), and 2) honest-but-curious (HBC) attackers that are part of the users of PPPML-HMI (type II) as in Definition 1.

Definition 1. The HBC attacker is a legitimate participant in a communication protocol who will not deviate from the defined protocol but will attempt to learn all possible information from legitimately received messages [95].

Sensitive information could be deciphered from medical images, such as tissue patterns and lesions, which could compromise patients' privacy [96]. CT scans of COVID-19 patients require even stronger privacy protection. To show that PPPML-HMI protected privacy from the CT scans of COVID-19 patients and resisted both types of attackers, we applied the iDLG (Method) to simulate an attacker reconstructing the

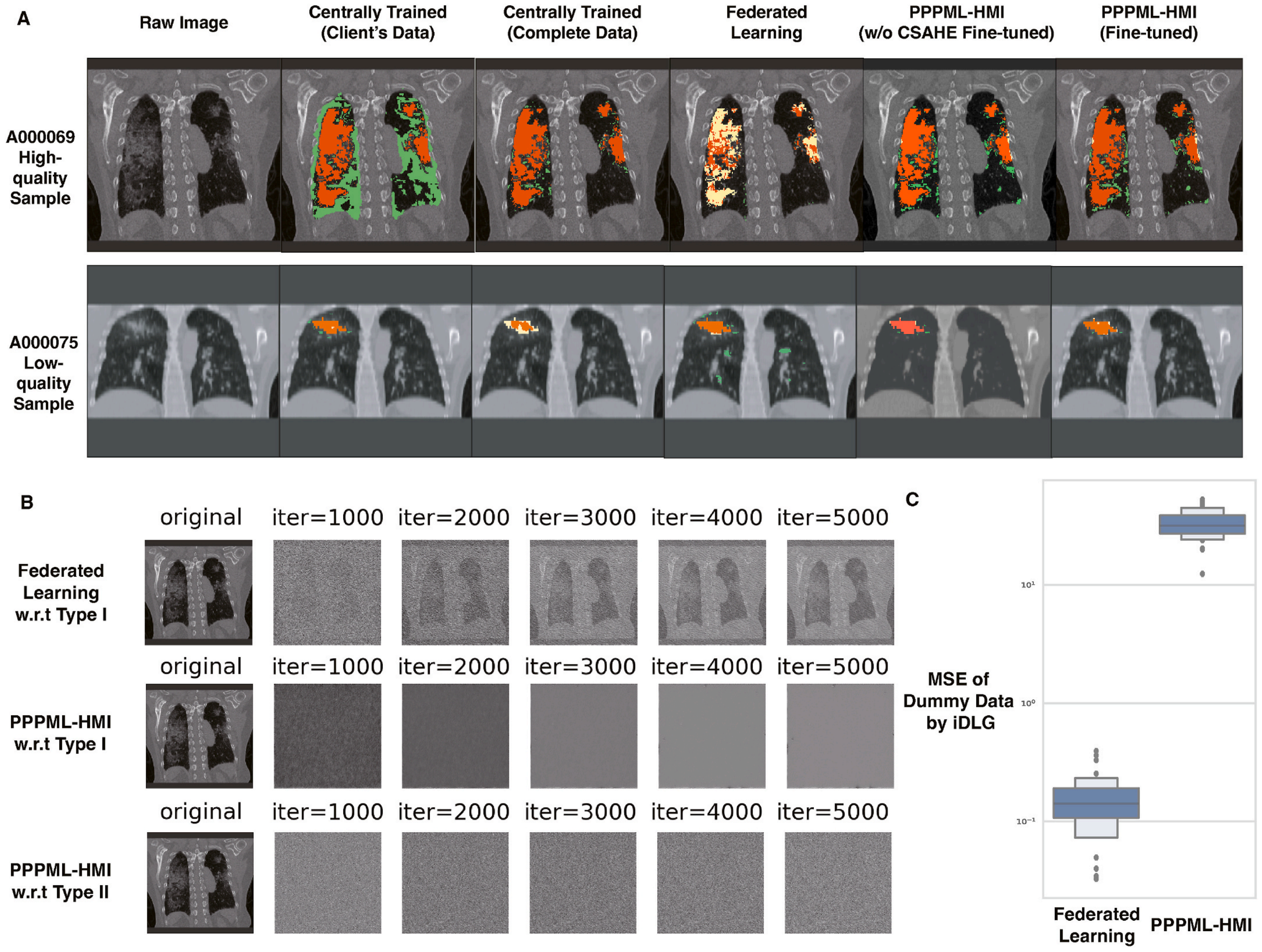

A
Raw Image
Centrally Trained (Client's Data)
Centrally Trained (Complete Data)
Federated Learning
PPPML-HMI (w/o CSAHE Fine-tuned)
PPPML-HMI (Fine-tuned)
A000069 High-quality Sample
A000075 Low-quality Sample
B
original
iter=1000
iter=2000
iter=3000
iter=4000
iter=5000
Federated Learning w.r.t Type I
PPPML-HMI w.r.t Type I
PPPML-HMI w.r.t Type II
C
MSE of Dummy Data by iDLG
10^{1}
10^{0}
10^{-1}
Federated Learning
PPPML-HMI

Fig. 4. A) Visualization of predicted segmentation mask on the high-quality sample A000069 and low-quality sample A000075 (Orange: true positives, Green: false positives, Yellow: false negatives). B) Visualization of the dummy data from the iDLG attack on FL and PPPML-HMI against both types of attackers. C) Distribution of MSE of the dummy data from the iDLG attack on FL and PPPML-HMI.

training images by stealing the gradients transmitted between users and the server. During each attack, we initialized dummy data and used the gradient transmitted between the user and server to update the dummy data. We performed the iDLG attack for FL, where type I attackers exist and PPPML-HMI, where both types of attackers exist, respectively. Then, we visualized the dummy data every 1000 iterations, as shown in Fig. 4B. To quantitatively assess the reconstructive capabilities of iDLG and illustrate the privacy-preserving capability of PPPML-HMI, we conducted experiments involving a random selection of 50 samples, which were subjected to reconstruction using iDLG within the Federated Learning (FL) and PPPML-HMI scenarios. Upon calculating the Mean Square Error (MSE) between the Dummy data and the real data as shown in Fig. 4C, our results indicated that iDLG could efficiently reconstruct the data in the FL scenario, exhibiting a low MSE. In contrast, PPPML-HMI demonstrates a substantially higher MSE, implying that PPPML-HMI blocked the reconstruction.

In practice, since the server is usually controlled by a third party, the users could not completely trust the server. As shown in Fig. 2B, in FL, type I attackers could intercept the gradients sent between users and the server to learn the sensitive information of the corresponding users. With the CSAHE, only the initiator could communicate with the server and send the securely aggregated gradient. Though type I attackers could intercept the gradient between the initiator and the server, they only get the averaged gradient over all users. When type I attackers incept the information transmitted between users in CSAHE, they only get the ciphertext instead of the plaintext as all gradients transmitted in the loop are encrypted with HE. Hence, type I attackers could be resisted by PPPML-HMI.

Type II attackers may exist in PPPML-HMI as gradients are transmitted between users as shown in Fig. 2B. For example, an HBC user in the loop may try to recover sensitive information using the gradient transmitted from the previous user due to curiosity. As the public key for HE is shared by all users in the loop and the private key is held only by the initiator, in the case of a private key leak, the HBC user could decrypt the gradient from the previous user and see the plaintext. However, the plaintext deciphered by any non-initiator users is always further protected by the Gaussian noise added by the initiator, thus type II attackers could be resisted as shown in Fig. 4B.

To demonstrate the results of gradient inversion, we applied iDLG to the associated segmentation model. As shown in Fig. 4B, in terms of visual results, the iDLG attack on FL could effectively reconstruct the CT images in the training data, while the iDLG attack on PPPML-HMI was effectively blocked. Overall, PPPML-HMI protected privacy by blocking the reconstruction of sensitive medical images.

4. Discussion

In this paper, we present a novel, robust and open-source method for personalized and privacy-preserving federated heterogeneous medical imaging analysis. PPPML-HMI is a training paradigm similar to FL, which has no task-specific requirements and does not require any modifications to the existing DL models. With the nature of open-source, users of PPPML-HMI only need to apply PPPML-HMI as a plug-in to their neural network models as how they work with FL to achieve personalized and privacy-preserving federated learning with even faster convergence speed (Fig. S1). Based on the results of the simulated classification task on the RAD-ChestCT dataset and the real-word segmentation task based on the COVID-19 dataset, we believe that PPPML-HMI could be applied to any potential medical imaging problem with different DL methods, especially for those with heterogeneous data and the need for federation and privacy protection, as the segmentation method we adopted in the real-world case was also a general method for segmentation of lung, tracheal, vascular and so on.

Though we applied PPPML-HMI in both simulated and real-world COVID-19 cases, all experiments were conducted in a laboratory environment, meaning all practical conditions were in the ideal state, including the computing power of the users' machines and the communication consumption between machines. As shown in Table 3, PPPML-HMI showed slightly higher requirements for the training time and similar memory and GPU compared to FL, meaning that PPPML-HMI could still work in the case that FL works. Meanwhile, integrating personalization and HE-based privacy protection in PPPML-HMI brought an additional 35.5 % computation time and higher memory storage requirements due to encryption and decryption compared to FL as shown in Table S1, thus finding new solutions to accelerate could be one of the main research directions in the future.

Due to the special design of PPPML-HMI, it can only be applied when the number of clients ≥3, meaning that PPPML-HMI is vulnerable when the number of clients equals 2. Suppose that the initiator is the type II attacker and there are two clients u_1 and u_2, then the initiator could decode the exact gradient of the other client and thus do the gradient inversion attack as shown in the classic FL setting by eliminating. Because u_1 holds $\Delta w_{k,\tau}^{CSA} = \Delta w_{k,\tau}^{1} + R$ and transfers $HE(\Delta w_{k,\tau}^{CSA})$ to u_2. Then, u_2 aggregates $\Delta w_{k,2}^{2}$ to $HE(\Delta w_{k,\tau}^{CSA})$ homomorphically. The aggregated gradients $HE(\Delta w_{k,\tau}^{CSA}) + HE(\Delta w_{k,\tau}^{2}) = HE(\Delta w_{k,\tau}^{1} + R) + HE(\Delta w_{k,\tau}^{2})$ goes back to the initiator u_1. In this case, eliminating happens when u_1 eliminates $HE(\Delta w_{k,\tau}^{1} + R)$ from the aggregated gradients and is able to know the value of $\Delta w_{k,\tau}^{2}$. Though PPPML-HMI is not exactly private for the case where there are only two clients, it could work as designed when the number of clients ≥3. Because the act of elimination becomes increasingly challenging when the number of clients surpasses three, and this difficulty further escalates as the client count rises. In practical scenarios, it's common to have a significantly greater number of hospitals as clients. Moreover, even in cases where only two clients are involved, the initiator is randomly chosen in each round. This random selection process ensures that there is no consistent gradient for one client to persistently eliminate another client, thereby reducing the susceptibility of the model to inversion attacks.

Nevertheless, regardless of the issue in computing resources and vulnerability, we need to take more practical problems into consideration, such as the imbalance of computing power among hospitals, the deviation of data quality of different hospitals, the latency in network communication between hospitals and the server, and so on. Those practical problems have not been addressed in this work as we were focusing on a learning paradigm. Still, we will solve those realistic obstacles and further improve PPPML-HMI in future work.

Table 3
Comparison of computation resource requirements on the COVID-19 task. n is the number of users. Virtual Memory, Physical Memory and GPU were measured as the averaged value across all users' machines.

Method	Training Time (hours)	Virtual Memory (GB)	Physical Memory (GB)	GPU (GB)
Centralized training	320.21	70.26	3.34	26.80
FL ($n = 5$)	110.37	54.91	3.36	26.80
FL + SA ($n = 5$)	156.12	55.36	3.36	26.80
PPPML-HMI ($n = 5$)	149.55	54.97	3.43	26.80

Data availability

For the classification task, we used the publicly available RAD-ChestCT Dataset [88,89]. For the segmentation task, the data from partner hospitals are available upon request.

Code availability

The code for this paper is publicly available at https://github.com/JoshuaChou2018/PPPML-HMI.

CRediT authorship contribution statement

Juexiao Zhou: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Longxi Zhou:** Formal analysis, Methodology, Software, Validation, Writing – review & editing. **Di Wang:** Conceptualization, Investigation, Writing – review & editing. **Xiaopeng Xu:** Investigation, Writing – review & editing. **Haoyang Li:** Validation, Writing – review & editing. **Yuetan Chu:** Writing – review & editing. **Wenkai Han:** Writing – review & editing. **Xin Gao:** Conceptualization, Funding acquisition, Investigation, Project administration, Resources, Supervision, Writing – review & editing.

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.

Acknowledgments

We acknowledge funding from King Abdullah University of Science and Technology (KAUST). Juexiao Zhou, Longxi Zhou, Xiaopeng Xu, Haoyang Li, Yuetan Chu and Xin Gao were supported in part by grants from the Office of Research Administration (ORA) at KAUST under award numbers FCC/1/1976-44-01, FCC/1/1976-45-01, URF/1/4352-01-01, REI/1/4742-01-01 REI/1/4473-01-01, and URF/1/4663-01-01. Di Wang was supported in part by the baseline funding BAS/1/1689-01-01 and funding from the AI Initiative REI/1/4811-10-01 of King Abdullah University of Science and Technology (KAUST).

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.compbiomed.2023.107861.

References

[1] Y. LeCun, Y. Bengio, G. Hinton, Deep Learn. Nat. 521 (2015) 436–444.

[2] H. Greenspan, B. Van Ginneken, R.M. Summers, Guest editorial deep learning in medical imaging: overview and future promise of an exciting new technique, IEEE Trans. Med. Imag. 35 (2016) 1153–1159.

[3] D.S. Ting, Y. Liu, P. Burlina, X. Xu, N.M. Bressler, T.Y. Wong, AI for medical imaging goes deep, Nat. Med. 24 (2018) 539–540.

[4] A.S. Lundervold, A. Lundervold, An overview of deep learning in medical imaging focusing on MRI, Z. Med. Phys. 29 (2019) 102–127.

[5] N. Rieke, J. Hancox, W. Li, F. Milletari, H.R. Roth, S. Albarqouni, S. Bakas, M. N. Galtier, B.A. Landman, K. Maier-Hein, The future of digital health with federated learning, NPJ Dig. Med. 3 (2020) 119.

[6] S.K. Zhou, H. Greenspan, C. Davatzikos, J.S. Duncan, B. Van Ginneken, A. Madabhushi, J.L. Prince, D. Rueckert, R.M. Summers, A review of deep learning in medical imaging: imaging traits, technology trends, case studies with progress highlights, and future promises, Proc. IEEE 109 (2021) 820–838.

[7] R. Aggarwal, V. Sounderajah, G. Martin, D.S. Ting, A. Karthikesalingam, D. King, H. Ashrafian, A. Darzi, Diagnostic accuracy of deep learning in medical imaging: a systematic review and meta-analysis, NPJ Dig. Med. 4 (2021) 65.

[8] L. Zhou, X. Meng, Y. Huang, K. Kang, J. Zhou, Y. Chu, H. Li, D. Xie, J. Zhang, W. Yang, An interpretable deep learning workflow for discovering subvisual abnormalities in CT scans of COVID-19 inpatients and survivors, Nat. Mach. Intell. 4 (2022) 494–503.

[9] J. Rasley, S. Rajbhandari, O. Ruwase, Y. He, Deepspeed: system optimizations enable training deep learning models with over 100 billion parameters, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 3505–3506.

[10] C. Sun, A. Shrivastava, S. Singh, A. Gupta, Revisiting unreasonable effectiveness of data in deep learning era, Proc. IEEE Int.Conf. Comput. Vis. (2017) 843–852.

[11] M.A. Nielsen, Neural Networks and Deep Learning, Determination press, San Francisco, CA, USA, 2015.

[12] F. Wang, L.P. Casalino, D. Khullar, Deep learning in medicine—promise, progress, and challenges, JAMA Intern. Med. 179 (2019) 293–294.

[13] P.M. Mammen, Federated Learning: Opportunities and Challenges, 2021 arXiv preprint arXiv:2101.05428.

[14] F. Shan, Y. Gao, J. Wang, W. Shi, N. Shi, M. Han, Z. Xue, D. Shen, Y. Shi, Lung Infection Quantification of COVID-19 in CT Images with Deep Learning, 2020 arXiv preprint arXiv:2003.04655.

[15] L. Zhou, Z. Li, J. Zhou, H. Li, Y. Chen, Y. Huang, D. Xie, L. Zhao, M. Fan, S. Hashmi, A rapid, accurate and machine-agnostic segmentation and quantification method for CT-based COVID-19 diagnosis, IEEE Trans. Med. Imag. 39 (2020) 2638–2652.

[16] S. Serte, H. Demirel, Deep learning for diagnosis of COVID-19 using 3D CT scans, Comput. Biol. Med. 132 (2021), 104306.

[17] A. Saood, I. Hatem, COVID-19 lung CT image segmentation using deep learning methods: U-Net versus SegNet, BMC Med. Imag. 21 (2021) 1–10.

[18] W.G. Van Panhuis, P. Paul, C. Emerson, J. Grefenstette, R. Wilder, A.J. Herbst, D. Heymann, D.S. Burke, A systematic review of barriers to data sharing in public health, BMC Publ. Health 14 (2014) 1–9.

[19] J. Konečný, H.B. McMahan, F.X. Yu, P. Richtárik, A.T. Suresh, D. Bacon, Federated Learning: Strategies for Improving Communication Efficiency, 2016 arXiv preprint arXiv:1610.05492.

[20] B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A. y Arcas, Communication-efficient Learning of Deep Networks from Decentralized Data, Artificial Intelligence and Statistics, PMLR, 2017, pp. 1273–1282.

[21] Q. Yang, Y. Liu, T. Chen, Y. Tong, Federated machine learning: concept and applications, ACM Transact. Intellig. Syst. Technol. (TIST) 10 (2019) 1–19.

[22] T. Li, A.K. Sahu, A. Talwalkar, V. Smith, Federated learning: challenges, methods, and future directions, IEEE Signal Process. Mag. 37 (2020) 50–60.

[23] M. Mohri, G. Sivek, A.T. Suresh, Agnostic Federated Learning, International Conference on Machine Learning, PMLR, 2019, pp. 4615–4625.

[24] D.J. Beutel, T. Topal, A. Mathur, X. Qiu, J. Fernandez-Marques, Y. Gao, L. Sani, K. H. Li, T. Parcollet, P.P.B. de Gusmão, Flower, A Friendly Federated Learning Research Framework, 2020 arXiv preprint arXiv:2007.14390.

[25] W. Zhuang, X. Gan, Y. Wen, S. Zhang, Easyfl: a low-code federated learning platform for dummies, IEEE Internet Things J. 9 (2022) 13740–13754.

[26] Y. Liu, T. Fan, T. Chen, Q. Xu, Q. Yang, Fate: an industrial grade platform for collaborative learning with data protection, J. Mach. Learn. Res. 22 (2021) 10320–10325.

[27] W. Zhang, F. Yu, X. Wang, X. Zeng, H. Zhao, Y. Tian, F.-Y. Wang, L. Li, Z. Li, R $ˆ {2} $ fed: resilient reinforcement federated learning for industrial applications, IEEE Trans. Ind. Inf. (2022).

[28] G. Mittone, N. Tonci, R. Birke, I. Colonnelli, D. Medić, A. Bartolini, R. Esposito, E. Parisi, F. Beneventi, M. Polato, Experimenting with Emerging ARM and RISC-V Systems for Decentralised Machine Learning, 2023 arXiv preprint arXiv: 2302.07946.

[29] G. Mittone, W. Riviera, I. Colonnelli, R. Birke, M. Aldinucci, Model-Agnostic Federated Learning, 2023 arXiv preprint arXiv:2303.04906.

[30] W. Zhang, Q. Lu, Q. Yu, Z. Li, Y. Liu, S.K. Lo, S. Chen, X. Xu, L. Zhu, Blockchain-based federated learning for device failure detection in industrial IoT, IEEE Internet Things J. 8 (2020) 5926–5937.

[31] S.P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, A.T. Suresh, Scaffold: stochastic controlled averaging for federated learning, in: International Conference on Machine Learning, PMLR, 2020, pp. 5132–5143.

[32] Y. Jiang, J. Konečný, K. Rush, S. Kannan, Improving Federated Learning Personalization via Model Agnostic Meta Learning, 2019 arXiv preprint arXiv: 1909.12488.

[33] F. Hanzely, S. Hanzely, S. Horváth, P. Richtárik, Lower bounds and optimal algorithms for personalized federated learning, Adv. Neural Inf. Process. Syst. 33 (2020) 2304–2315.

[34] A. Fallah, A. Mokhtari, A. Ozdaglar, Personalized Federated Learning: A Meta-Learning Approach, 2020 arXiv preprint arXiv:2002.07948.

[35] C. T Dinh, N. Tran, J. Nguyen, Personalized federated learning with moreau envelopes, Adv. Neural Inf. Process. Syst. 33 (2020) 21394–21405.

[36] Y. Mansour, M. Mohri, J. Ro, A.T. Suresh, Three Approaches for Personalization with Applications to Federated Learning, 2020 arXiv preprint arXiv:2002.10619.

[37] P. Sun, H. Che, Z. Wang, Y. Wang, T. Wang, L. Wu, H. Shao, Pain-FL: personalized privacy-preserving incentive for federated learning, IEEE J. Sel. Area. Commun. 39 (2021) 3805–3820.

[38] T. Li, S. Hu, A. Beirami, V. Smith, Ditto: fair and robust federated learning through personalization, in: International Conference on Machine Learning, PMLR, 2021, pp. 6357–6368.

[39] P. Kairouz, H.B. McMahan, B. Avent, A. Bellet, M. Bennis, A.N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, Advances and open problems in federated learning, Found. Trends® Mach. Learn. 14 (2021) 1–210.

[40] Y. Deng, M.M. Kamani, M. Mahdavi, Adaptive Personalized Federated Learning, 2020 arXiv preprint arXiv:2003.13461.

[41] J. Wang, Y. Jin, L. Wang, Personalizing Federated Medical Image Segmentation via Local Calibration, European Conference on Computer Vision, Springer, 2022, pp. 456–472.

[42] Z. Gao, L. Li, F. Wu, S. Wang, X. Zhuang, Decoupling predictions in distributed learning for multi-center left atrial MRI segmentation. International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2022, pp. 517–527.

[43] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, V. Chandra, Federated Learning with Non-iid Data, 2018 arXiv preprint arXiv:1806.00582.

[44] L. Qu, N. Balachandar, M. Zhang, D. Rubin, Handling data heterogeneity with generative replay in collaborative learning for medical imaging, Med. Image Anal. 78 (2022), 102424.

[45] C. Wu, F. Wu, L. Lyu, T. Qi, Y. Huang, X. Xie, A federated graph neural network framework for privacy-preserving personalization, Nat. Commun. 13 (2022) 3091.

[46] W.N. Price, I.G. Cohen, Privacy in the age of medical big data, Nat. Med. 25 (2019) 37–43.

[47] V. Tolpegin, S. Truex, M.E. Gursoy, L. Liu, in: Data Poisoning Attacks against Federated Learning Systems, Computer Security–ESORICS 2020: 25th European Symposium on Research in Computer Security, ESORICS 2020, Springer, Guildford, UK, 2020, pp. 480–501. September 14–18, 2020, Proceedings, Part I 25.

[48] M.A. Rahman, T. Rahman, R. Laganière, N. Mohammed, Y. Wang, Membership inference attack against differentially private deep learning model, Trans. Data Priv. 11 (2018) 61–79.

[49] H. Lu, C. Liu, T. He, S. Wang, K.S. Chan, Sharing Models or Coresets: A Study Based on Membership Inference Attack, 2020 arXiv preprint arXiv:2007.02977.

[50] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, M. Backes, Ml-leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models, 2018 arXiv preprint arXiv:1806.01246.

[51] H. Hu, Z. Salcic, L. Sun, G. Dobbie, X. Zhang, Source inference attacks in federated learning, in: 2021 IEEE International Conference on Data Mining (ICDM), IEEE, 2021, pp. 1102–1107.

[52] L. Lyu, C. Chen, A Novel Attribute Reconstruction Attack in Federated Learning, 2021 arXiv preprint arXiv:2108.06910.

[53] J. Geiping, H. Bauermeister, H. Dröge, M. Moeller, Inverting gradients-how easy is it to break privacy in federated learning? Adv. Neural Inf. Process. Syst. 33 (2020) 16937–16947.

[54] B. Zhao, K.R. Mopuri, H. Bilen, idlg, Improved Deep Leakage from Gradients, 2020 arXiv preprint arXiv:2001.02610.

[55] H. Yin, A. Mallya, A. Vahdat, J.M. Alvarez, J. Kautz, P. Molchanov, See through gradients: image batch recovery via gradinversion, Proc. IEEE/CVF Conf. Comput. Vis.Pattern Recog. (2021) 16337–16346.

[56] A. Hatamizadeh, H. Yin, P. Molchanov, A. Myronenko, W. Li, P. Dogra, A. Feng, M. G. Flores, J. Kautz, D. Xu, Do gradient inversion attacks make federated learning unsafe? IEEE Trans. Med. Imag. (2023).

[57] A. Lalitha, S. Shekhar, T. Javidi, F. Koushanfar, Fully Decentralized Federated Learning, Third Workshop on Bayesian Deep Learning, NeurIPS), 2018.

[58] L. Yuan, L. Sun, P.S. Yu, Z. Wang, Decentralized Federated Learning: A Survey and Perspective, 2023 arXiv preprint arXiv:2306.01603.

[59] A. Lalitha, O.C. Kilinc, T. Javidi, F. Koushanfar, Peer-to-peer Federated Learning on Graphs, 2019 arXiv preprint arXiv:1901.11173.

[60] C. He, C. Tan, H. Tang, S. Qiu, J. Liu, Central Server Free Federated Learning over Single-Sided Trust Social Networks, 2019 arXiv preprint arXiv:1910.04956.

[61] C. He, E. Ceyani, K. Balasubramanian, M. Annavaram, S. Avestimehr, Spreadgnn: Serverless Multi-Task Federated Learning for Graph Neural Networks, 2021 arXiv preprint arXiv:2106.02743.

[62] H. Xing, O. Simeone, S. Bi, Federated learning over wireless device-to-device networks: algorithms and convergence analysis, IEEE J. Sel. Area. Commun. 39 (2021) 3723–3741.

[63] S. Warnat-Herresthal, H. Schultze, K.L. Shastry, S. Manamohan, S. Mukherjee, V. Garg, R. Sarveswara, K. Händler, P. Pickkers, N.A. Aziz, Swarm learning for decentralized and confidential clinical machine learning, Nature 594 (2021) 265–270.

[64] S. Kalra, J. Wen, J.C. Cresswell, M. Volkovs, H. Tizhoosh, Decentralized federated learning through proxy model sharing, Nat. Commun. 14 (2023) 2899.

[65] C. Dwork, Differential Privacy: A Survey of Results, International Conference on Theory and Applications of Models of Computation, Springer, 2008, pp. 1–19.

[66] R. Hu, Y. Guo, H. Li, Q. Pei, Y. Gong, Personalized federated learning with differential privacy, IEEE Internet Things J. 7 (2020) 9530–9539.

[67] J. Zhou, S. Chen, Y. Wu, H. Li, B. Zhang, L. Zhou, Y. Hu, Z. Xiang, Z. Li, N. Chen, PPML-omics: a Privacy-Preserving Federated Machine Learning Method Protects Patients' Privacy in Omic Data, bioRxiv, 2022, 2022.2003. 2023.485485.

[68] E. Bagdasaryan, O. Poursaeed, V. Shmatikov, Differential privacy has disparate impact on model accuracy, Adv. Neural Inf. Process. Syst. (2019) 32.

[69] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H.B. McMahan, S. Patel, D. Ramage, A. Segal, K. Seth, Practical secure aggregation for privacy-preserving machine learning. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 2017, pp. 1175–1191.

[70] C. Gentry, Fully homomorphic encryption using ideal lattices. Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing, 2009, pp. 169–178.

[71] O. Goldreich, Secure Multi-Party Computation, Manuscript, Preliminary version, 1998, p. 78.

[72] T. Sandholm, S. Mukherjee, B.A. Huberman, SAFE: Secure Aggregation with Failover and Encryption, 2021 arXiv preprint arXiv:2108.05475.

[73] A. Ziller, A. Trask, A. Lopardo, B. Szymkow, B. Wagner, E. Bluemke, J.-M. Nounahon, J. Passerat-Palmbach, K. Prakash, N. Rose, Pysyft: a library for easy federated learning, Federat. Learn. Syst.: Towards Next-Generat. AI (2021) 111–139.

[74] H.R. Roth, Y. Cheng, Y. Wen, I. Yang, Z. Xu, Y.-T. Hsieh, K. Kersten, A. Harouni, C. Zhao, K. Lu, Nvidia Flare: Federated Learning from Simulation to Real-World, 2022 arXiv preprint arXiv:2210.13291.

[75] A.S. Fauci, H.C. Lane, R.R. Redfield, Covid-19—navigating the Uncharted, Mass Medical Soc, 2020, pp. 1268–1269.

[76] T. Franquet, Imaging of pneumonia: trends and algorithms, Eur. Respir. J. 18 (2001) 196–208.

[77] S. Wang, Y. Zha, W. Li, Q. Wu, X. Li, M. Niu, M. Wang, X. Qiu, H. Li, H. Yu, A fully automatic deep learning system for COVID-19 diagnostic and prognostic analysis, Eur. Respir. J. 56 (2020).

[78] M. Jamshidi, A. Lalbakhsh, J. Talla, Z. Peroutka, F. Hadjilooei, P. Lalbakhsh, M. Jamshidi, L. La Spada, M. Mirmozafari, M. Dehghani, Artificial intelligence and COVID-19: deep learning approaches for diagnosis and treatment, IEEE Access 8 (2020) 109581–109595.

[79] C. Shorten, T.M. Khoshgoftaar, B. Furht, Deep learning applications for COVID-19, J. Big Data 8 (2021) 1–54.

[80] A.M. Ismael, A. Şengür, Deep learning approaches for COVID-19 detection based on chest X-ray images, Expert Syst. Appl. 164 (2021), 114054.

[81] N. Subramanian, O. Elharrouss, S. Al-Maadeed, M. Chowdhury, A review of deep learning-based detection methods for COVID-19, Comput. Biol. Med. 143 (2022), 105233.

[82] I. Dayan, H.R. Roth, A. Zhong, A. Harouni, A. Gentili, A.Z. Abidin, A. Liu, A. B. Costa, B.J. Wood, C.-S. Tsai, Federated learning for predicting clinical outcomes in patients with COVID-19, Nat. Med. 27 (2021) 1735–1743.

[83] R. Durga, E. Poovammal, Fled-block: federated learning ensembled deep learning blockchain model for covid-19 prediction, Front. Public Health 10 (2022), 892499.

[84] O. Samuel, A.B. Omojo, A.M. Onuja, Y. Sunday, P. Tiwari, D. Gupta, G. Hafeez, A. S. Yahaya, O.J. Fatoba, S. Shamshirband, IoMT: a COVID-19 healthcare system driven by federated learning and blockchain, IEEE J. Biomed.Health Inform. 27 (2022) 823–834.

[85] L.M. Florescu, C.T. Streba, M.-S. Şerbănescu, M. Mămuleanu, D.N. Florescu, R. V. Teică, R.E. Nica, I.A. Gheonea, Federated learning approach with pre-trained deep learning models for covid-19 detection from unsegmented ct images, Life 12 (2022) 958.

[86] Z. Li, X. Xu, X. Cao, W. Liu, Y. Zhang, D. Chen, H. Dai, Integrated CNN and federated learning for COVID-19 detection on chest X-ray images, IEEE ACM Trans. Comput. Biol. Bioinf (2022).

[87] F. Wibawa, F.O. Catak, M. Kuzlu, S. Sarp, U. Cali, Homomorphic encryption and federated learning based privacy-preserving cnn training: covid-19 detection use-case, in: Proceedings of the 2022 European Interdisciplinary Cybersecurity Conference, 2022, pp. 85–90.

[88] R.L. Draelos, D. Dov, M.A. Mazurowski, J.Y. Lo, R. Henao, G.D. Rubin, L. Carin, Machine-learning-based multiple abnormality prediction with large-scale chest computed tomography volumes, Med. Image Anal. 67 (2021), 101857.

[89] R.L. Draelos, D. Dov, M.A. Mazurowski, J.Y. Lo, R. Henao, G.D. Rubin, L. Carin, RAD-ChestCT Dataset, 2020. Zenodo.

[90] T.D. Bui, J. Shin, T. Moon, 3D Densely Convolutional Networks for Volumetric Segmentation, 2017 arXiv preprint arXiv:1709.03199.

[91] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 1126–1135.

[92] A. Shamir, How to share a secret, Commun. ACM 22 (1979) 612–613.

[93] J.H. Cheon, A. Kim, M. Kim, Y. Song, Homomorphic encryption for arithmetic of approximate numbers, in: Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, pp. 409–437, 2017, Proceedings, Part I 23, Springer.

[94] A. Benaissa, B. Retiat, B. Cebere, A.E. Belfedhal, TenSEAL, A Library for Encrypted Tensor Operations Using Homomorphic Encryption, 2021 arXiv preprint arXiv:2104.03152.

[95] A. Paverd, A. Martin, I. Brown, Modelling and automatically analysing privacy properties for honest-but-curious adversaries, Tech. Rep. (2014).

[96] G.A. Kaissis, M.R. Makowski, D. Rückert, R.F. Braren, Secure, privacy-preserving and federated machine learning in medical imaging, Nat. Mach. Intell. 2 (2020) 305–311.