FedHealth 2: Weighted Federated Transfer Learning via Batch Normalization for Personalized Healthcare

Yiqiang Chen[1,2*], **Wang Lu**[1,2], **Jindong Wang**[3] and **Xin Qin**[1,2]
[1]Institute of Computing Technology, Chinese Academy of Sciences, 100190, Beijing, China
[2]University of Chinese Academy of Sciences, 100190, Beijing, China
[3]Microsoft Research Asia, Beijing, China
{yqchen,luwang,qinxin18b}@ict.ac.cn, jindong.wang@microsoft.com

Abstract

The success of machine learning applications often needs a large quantity of data. Recently, federated learning (FL) is attracting increasing attention due to the demand for data privacy and security, especially in the medical field. However, the performance of existing FL approaches often deteriorate when there exist domain shifts among clients, and few previous works focus on personalization in healthcare. In this article, we propose FedHealth 2, an extension of FedHealth [Chen *et al.*, 2020] to tackle domain shifts and get personalized models for local clients. FedHealth 2 obtains the client similarities via a pretrained model, and then it averages all weighted models with preserving local batch normalization. Wearable activity recognition and COVID-19 auxiliary diagnosis experiments have evaluated that FedHealth 2 can achieve better accuracy (**10**%+ improvement for activity recognition) and personalized healthcare without compromising privacy and security.

1 Introduction

For the past few years, machine learning technology has been widely used in real life. Machine learning technology leverages the labeled data to learn the patterns and trends of observed objects. With the help of perception technology, well-trained machine learning models liberate people from heavy and repetitive tasks, especially in the healthcare field. A successful machine learning healthcare application often requires sufficient user data to dig into the essence of health status. However, with the increasing awareness of privacy and security of the people and organizations, China, the European Union, and some other governments or organizations enforce the protection of user data via different regulations [Inkster, 2018; Voigt and Von dem Bussche, 2017]. In this situation, federated learning [Yang *et al.*, 2019] came into being to protect data privacy and security.

The most classic algorithm in FL is FedAvg [McMahan *et al.*, 2017], which directly averages the parameters of models coming from all participating clients without transporting local data. Although FedAvg is simple, it has demonstrated superior performance in many situations. However, FedAvg is not designed particularly for non-iid data, which may make it suffer from performance degradation or even diverge when it is deployed over non-iid samples [Li *et al.*, 2019]. As shown in Figure 1, different clients (users) cannot exchange their own data straightforward, and they exchange their information via model parameters with the help of the cloud. Various distributions of the client data make it hard to achieve good performance when directly employing FedAvg. The previous work, FedHealth [Chen *et al.*, 2020], is the first federated transfer learning framework for wearable healthcare. Although it achieves great success in federated learning for healthcare and solves the problems of data islanding and personalization, it requires a large number of data shared across different clients, which is unrealistic in many situations.

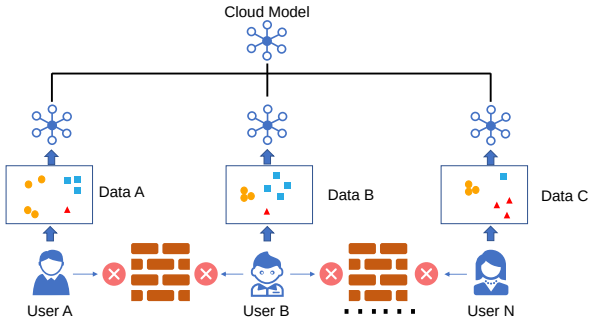


Figure 1: Data islanding and personalization in healthcare.

In this article, we propose FedHealth 2, a weighted federated transfer learning algorithm via batch normalization for personalized healthcare. FedHealth 2 can solve both data islanding and personalization problems without sharing common data. Specifically, FedHealth 2 gets the similarities among clients with the help of a pretrained model. The similarities are determined by the distances of the data distributions, and the distances can be calculated via the statistical values of the layers' outputs of the pretrained network. After obtaining the similarities, the cloud averages the weighted models' parameters in a personalized way and generates a unique model for each client. Each client preserves its own

*Corresponding Author

batch normalization and updates the model with a momentum method. Obviously, FedHealth 2 can not only cope with feature shift non-iid [Li *et al.*, 2021] via keeping local batch normalization but also cope with some other shifts, such as label shifts by considering clients' similarities. In addition, all parameter exchange processes are encrypted to ensure that data is not leaked. FedHealth 2 is extensible and can be deployed to many healthcare applications with different expansion methods.

Our contributions are as follows.

1. We propose FedHealth 2, a weighted federated transfer learning algorithm via batch normalization for healthcare, which can aggregate the information from different clients without compromising privacy security, and achieve personalized models for clients through weighting models and preserving local batch normalization.
2. We show the excellent performance achieved by FedHealth 2 in healthcare applications. Experiments show that FedHealth 2 dramatically improves the recognition accuracy of the local clients' models. At the same time, it reduces the number of rounds and speeds up the convergence to some extent.
3. FedHealth 2 is extensible and can be employed in many healthcare applications. Diverse extensions allow it can work in many circumstances. Even if there does not exist a pretrained model, FedHealth 2 can get similarities via FedBN [Li *et al.*, 2021] with few rounds.

2 Related Work

2.1 Machine Learning and Healthcare

With the rapid development of the technology of perception and computing, people can make use of machine learning to help doctors diagnose, assist doctors in the operation, etc. And with the help of perception technology, machine learning even can carry out disease warnings via daily behavior supervision. For instance, certain activities in daily life reflect early signals of some cognitive disease. Through daily observation of gait changes and finger flexibility, the machine can tell people whether they are suffering from Parkinson [Chen *et al.*, 2017].

Unfortunately, a successful healthcare application needs a large amount of labeled data of users. However, in real applications, data are often separate and few people are willing to disclose their private data. In addition, an increasing number of regulations, such as [Inkster, 2018; Voigt and Von dem Bussche, 2017], hold back the leakages of data.

2.2 Federated Learning

Federated learning is a usual way to combine each client's information while protecting data privacy and security. It was first proposed by Google [McMahan *et al.*, 2017], where they proposed FedAvg to train machine learning models via aggregating distributed mobile phones' information without exchanging data. The key idea is to replace direct data exchanges with model parameter-related exchanges. FedAvg is able to resolve the data islanding problems.

Though FedAvg works well in many situations, it still suffers from performance degradation or even diverges when meeting non-iid issues. Some works try to tackle these problems. FedProx [Li *et al.*, 2018] tackled the heterogeneity by allowing partial information aggregation and adding a proximal term to FedAvg. [Yeganeh *et al.*, 2020] aggregated the models of the clients with weights computed via L_1 distance among client models' parameters. These works focus on a common model shared by all clients while some other works try to obtain a unique model for each client. [Arivazhagan *et al.*, 2019] exchanged base layers' information and preserved personalization layer to combat the ill-effects of heterogeneity. [Dinh *et al.*, 2020] utilized Moreau envelopes as clients' regularized loss function and decoupled personalized model optimization from the global model learning in a bi-level problem stylized for personalized FL. [Yu *et al.*, 2020] evaluated three techniques for local adaptation of federated models: fine-tuning, multi-task learning, and knowledge distillation. Two works most relevant to our method are FedHealth [Chen *et al.*, 2020] and FedBN [Li *et al.*, 2021]. FedHealth proposed a federated transfer learning framework which needs some sharing data in healthcare while FedBN used local batch normalization to alleviate the feature shift before averaging models. Although there are already some works to cope with non-iid issues, few works pay attention to feature shift non-iid and other shifts at the same time and getting individual model for each client in healthcare.

2.3 Batch Normalization

Batch Normalization (BN) [Ioffe and Szegedy, 2015] has been an indispensable component of deep learning since it was created. Batch Normalization improves the performance of the model and has a natural advantage in dealing with domain shifts. Nowadays, researchers have explored many effects of BN, especially in transfer learning [Segù *et al.*, 2020]. FedBN [Li *et al.*, 2021] is one of few applications of BN in the field of FL field. However, FedBN does still not make full use of BN properties, and it does not consider the similarities among the clients.

3 Method

3.1 Problem Formulation

In a FL problem, there are N different clients (organizations or users), denoted as $\{C_1, C_2, \cdots, C_N\}$ and each client has its own dataset, i.e. $\{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_N\}$. Each dataset, $\mathcal{D}_i = \{(\mathbf{x}_{i,j}, y_{i,j})\}_{j=1}^{n_i}$, contains two parts, i.e. a train dataset $\mathcal{D}_i^{train} = \{(\mathbf{x}_{i,j}^{train}, y_{i,j}^{train})\}_{j=1}^{n_i^{train}}$ and a test dataset $\mathcal{D}_i^{test} = \{(\mathbf{x}_{i,j}^{test}, y_{i,j}^{test})\}_{j=1}^{n_i^{test}}$. Obviously, we have $n_i = n_i^{train} + n_i^{test}$ and $\mathcal{D}_i = \mathcal{D}_i^{train} \cup \mathcal{D}_i^{test}$. All of the datasets have different distributions, i.e. $P(\mathcal{D}_i) \neq P(\mathcal{D}_j)$. Each client has its own model denoted as $\{f_i\}_{i=1}^N$. Our goal is to combine information of all clients to learn a good model f_i for each client on its local dataset $\mathcal{D}_i$ without private data leakage:

$$\min_{\{f_k\}_{k=1}^N} \frac{1}{N} \sum_{i=1}^{N} \frac{1}{n_i^{test}} \sum_{j=1}^{n_i^{test}} \ell(f_i(\mathbf{x}_{i,j}^{test}), y_{i,j}^{test}), \qquad (1)$$

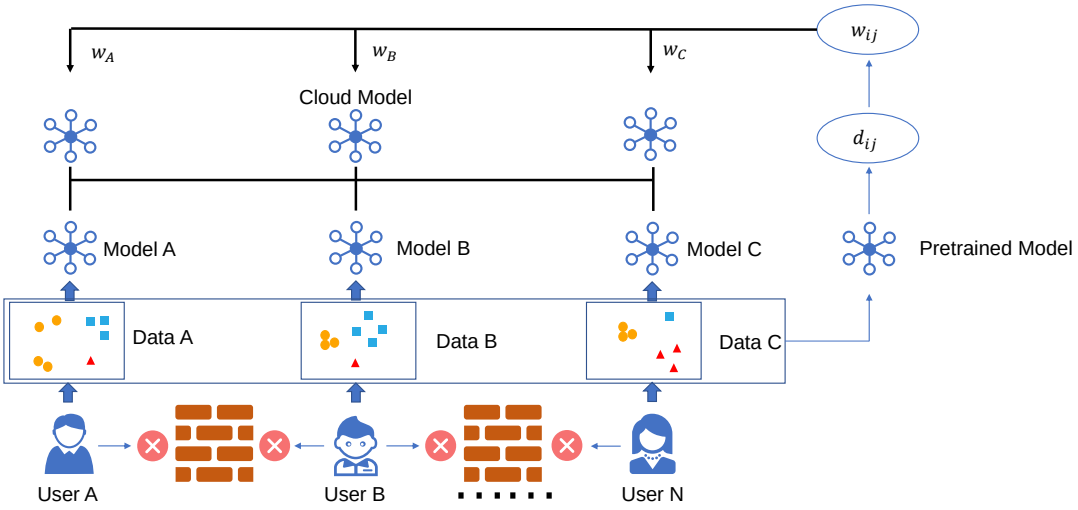


Figure 2: The overall structure of the FedHealth 2 method for federated learning.

where ℓ is a loss function.

3.2 Overview of FedHealth 2

FedHealth 2 aims to achieve accurate personal healthcare through weighted federated transfer learning via local batch normalization without compromising data privacy and security. Figure 2 gives an overview of the structure. Without loss of generality, we assume there are three clients, which can be extended to the more general case. The structure mainly contains five parts. Firstly, the server distributes the pretrained model to each client. Secondly, each client computes statistics of the outputs of specific layers according to local data. Thirdly, the server obtains clients' similarities represented by the weight matrix $\mathbf{W}$ which will be used to guide aggregation. Fourthly, each client updates its own model with the local train data and pushes their models to the cloud. Finally, the server aggregates models and obtains N models delivered to N clients respectively. All processes do not involve the direct transmission of data, so FedHealth 2 avoids the leakage of data privacy to a certain extent.

The keys of FedHealth 2 are obtaining $\mathbf{W}$ and aggregating the models. The way of computing $\mathbf{W}$ will be introduced in the following, and we first describe how to aggregate the models after obtaining $\mathbf{W}$.

We denote the parameters of each model f_i as $\boldsymbol{\theta}_i = \boldsymbol{\phi}_i \cup \boldsymbol{\psi}_i$. $\boldsymbol{\phi}_i$ corresponds to the parameters of BN layers while $\boldsymbol{\psi}_i$ corresponds to the parameters of the other layers. $\mathbf{W}$ is an $N \times N$ matrix, which describes the similarities among the clients. w_{ij} demonstrates the similarity between client i and client j. The bigger w_{ij} is, the more similar the two clients are. We have $w_{ii} = \lambda$, where λ is a hyper-parameter, and $\sum_{j=1}^{N} w_{ij} = 1$.

Figure 3 demonstrates the process of aggregating the models. As shown in Figure 3, $\boldsymbol{\phi}_i$ is fixed while $\boldsymbol{\psi}_i$ is computed according to $\mathbf{w}_i$, where $\mathbf{w}_i$ means the ith rows of $\mathbf{W}$, and $\boldsymbol{\psi}$, where $\boldsymbol{\psi} = \{\boldsymbol{\psi}_i\}_{i=1}^{N}$. Let $\boldsymbol{\theta}_i^t = \boldsymbol{\phi}_i^t \cup \boldsymbol{\psi}_i^t$ represents the parameters of the model form client i in the round t, then we have

$$\begin{cases} \boldsymbol{\phi}_i^{t+1} = & \boldsymbol{\phi}_i^t \\ \boldsymbol{\psi}_i^{t+1} = & \sum_{j=1}^{N} w_{ij} \boldsymbol{\psi}_j^t. \end{cases} \tag{2}$$

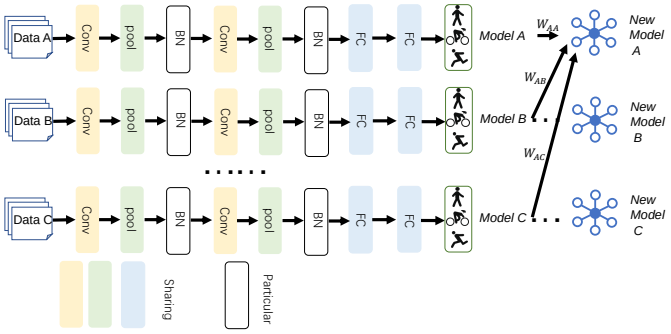


Figure 3: The concrete process of the FedHealth 2.

The overall process of FedHealth 2 is described in Algorithm 1. In next sections, we will introduce how to compute the weight matrix $\mathbf{W}$ with or without a pretrained model.

3.3 Evaluate Weights With a Pretrained Model

In this section, we will evaluate the weights with a pretrained model f and propose two ways to compute the weights.

We denote with a $l \in \{1, 2, \cdots, L\}$ in superscript notations the different batch normalization layers in the model. And $\mathbf{z}^{i,l}$ represents the input of lth batch normalization layer in the ith client. The input of the classify layer in the ith client is denoted as $\mathbf{z}^i$ which represents the domain features. We assume $\mathbf{z}^{i,l}$ is a matrix, $\mathbf{z}^{i,l}_{c_{i,l} \times s_{i,l}}$ where $c_{i,l}$ corresponds to the channel number while $s_{i,l}$ is the product of the other dimensions. Similarly, $\mathbf{z}^i = \mathbf{z}^i_{c_i \times s_i}$. We feed $\mathcal{D}_i$ into f, and we can obtain $\mathbf{z}^{i,l}_{c_{i,l} \times s_{i,l}}$. Obviously, $s_{i,l} = e \times n_i$ where e is an integer. Now, we try to compute statistics on the channels, and we treat $\mathbf{z}^{i,l}$ as a Gaussian distribution. For the lth layer

Algorithm 1 FedHealth 2

Input: A pretrained model f, N clients' datasets $\{\mathcal{D}_i\}_{i=1}^N$, λ
Output: Client models $\{f_i\}_{i=1}^N$

1: Distribute f to each client
2: Each client computes its statistics $(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i)$, where $\boldsymbol{\mu}_i$ represents the mean values while $\boldsymbol{\sigma}_i$ represents the covariance matrixs. Push $(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i)$ to the cloud
3: Compute $\mathbf{W}$ according to the statistics
4: Update the client model with local data. Push $\{\boldsymbol{\theta}_i^t\}_{i=1}^N$ to the cloud
5: Update $\{\boldsymbol{\theta}_i^t\}_{i=1}^N$ according to Eq. 2 and distribute $\{\boldsymbol{\theta}_i^{t+1}\}_{i=1}^N$ to the corresponding clients
6: Repeat steps $4 \sim 5$ util convergence or maximum round

of the ith client, it is easy to obtain its distribution parameters, $\mathcal{N}(\boldsymbol{\mu}^{i,l}, \boldsymbol{\sigma}^{i,l})$. And the ith client's statitics are denoted as

$$(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i) = [(\boldsymbol{\mu}^{i,1}, \boldsymbol{\sigma}^{i,1}), (\boldsymbol{\mu}^{i,2}, \boldsymbol{\sigma}^{i,2}), \cdots, (\boldsymbol{\mu}^{i,L}, \boldsymbol{\sigma}^{i,L})]. \tag{3}$$

Now we can calculate the distance between two clients. We usually utilize Wasserstein distance to calculate the distance between two Gaussian distributions. According to [Peyré *et al.*, 2019],

$$\begin{aligned} &W_2^2(\mathcal{N}(\boldsymbol{\mu}^{i,l}, \boldsymbol{\sigma}^{i,l}), \mathcal{N}(\boldsymbol{\mu}^{j,l}, \boldsymbol{\sigma}^{j,l})) \\ =&||\boldsymbol{\mu}^{i,l} - \boldsymbol{\mu}^{j,l}||^2 + \\ &tr(\boldsymbol{\sigma}^{i,l} + \boldsymbol{\sigma}^{j,l} - 2((\boldsymbol{\sigma}^{i,l})^{1/2}\boldsymbol{\sigma}^{j,l}(\boldsymbol{\sigma}^{i,l})^{1/2})^{1/2}), \end{aligned} \tag{4}$$

where tr is the trace of the matrix. Obviously, it is too difficulty to perform efficient calculations. Similar to BN, we perform approximations and consider that each channel is independent of each other. Therefore, $\boldsymbol{\sigma}^{i,l}$ is a diagonal matrix, i.e. $\boldsymbol{\sigma}^{i,l} = Diag(\mathbf{r}^{i,l})$. Now, we have

$$\begin{aligned} &W_2^2(\mathcal{N}(\boldsymbol{\mu}^{i,l}, \boldsymbol{\sigma}^{i,l}), \mathcal{N}(\boldsymbol{\mu}^{j,l}, \boldsymbol{\sigma}^{j,l})) \\ =&||\boldsymbol{\mu}^{i,l} - \boldsymbol{\mu}^{j,l}||^2 + ||\sqrt{\mathbf{r}^{i,l}} - \sqrt{\mathbf{r}^{j,l}}||_2^2. \end{aligned} \tag{5}$$

Therefore, we have

$$\begin{aligned} d_{i,j} &= \sum_{l=1}^{L} W_2(\mathcal{N}(\boldsymbol{\mu}^{i,l}, \boldsymbol{\sigma}^{i,l}), \mathcal{N}(\boldsymbol{\mu}^{j,l}, \boldsymbol{\sigma}^{j,l})) \\ &= \sum_{l=1}^{L} (||\boldsymbol{\mu}^{i,l} - \boldsymbol{\mu}^{j,l}||^2 + ||\sqrt{\mathbf{r}^{i,l}} - \sqrt{\mathbf{r}^{j,l}}||_2^2)^{1/2}. \end{aligned} \tag{6}$$

Big $d_{i,j}$ means the distribution distance between the ith client and the jth client is large. Therefore, the bigger $d_{i,j}$ is, the less similar two clients are, which means the smaller $w_{i,j}$ is. And we set $\tilde{w}_{i,j}$ as the inverse of $d_{i,j}$, i.e. $\tilde{w}_{i,j} = 1/d_{i,j}$. Since, $d_{i,i} = 0$, we let $w_{i,i} = \lambda$ and $\sum_{j=1, j\neq i}^{N} w_{i,j} = 1 - \lambda$. Normalize $\tilde{w}_i$ and we have

$$w_{i,j} = \begin{cases} \lambda, & j = i \\ (1-\lambda) \times \frac{\tilde{w}_{i,j}}{\sum_{j=1, j\neq i}^{N} \tilde{w}_{i,j}}, & j \neq i \end{cases} \tag{7}$$

We denote this weighting method as FedHealth 2. Similarly, for $\mathbf{z}^i$, we can obtain the corresponding $\mathbf{W}$ and we denote this method as d-FedHealth 2.

3.4 Evaluate Weights Without a Pretrained Model

Sometimes, there does not exist a pretrained model. In this situation, we can evaluate weights with models trained from several rounds of FedBN.

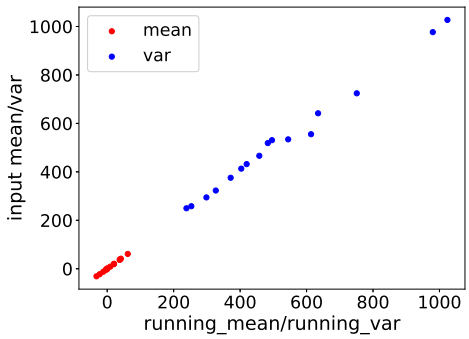
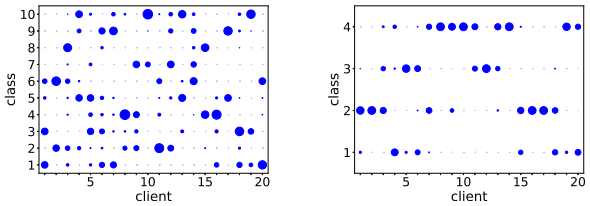


Figure 4: The running mean, running var of a BN layer and the inputs statistics of the corresponding layer in a client model.

As we can see from Figure 4, the running mean of the BN layer has a positive correlation with the statistical mean of the corresponding layer's inputs. And the variance has a similar relationship. From this, we can use running means and running variances of the BN layers instead of the statistics respectively. Therefore, we can perform several rounds of FedBN [Li *et al.*, 2021] which preserves local batch normalization, and utilize parameters of BN layers replacing the statistics when there does not exist a pretrained model. We denote this extension as f-FedHealth 2.

4 Experiments

4.1 Datasets



(a) PAMAP (b) COVID-19

Figure 5: The number of samples per class allocated to each client (indicated by dot size).

PAMAP. We adopt a public human activity recognition dataset called PAMAP [Reiss and Stricker, 2012]. The PAMAP dataset contains data of 18 different physical activities, performed by 9 subjects wearing 3 inertial measurement units and a heart rate monitor. We use data of 3 inertial measurement units which are collected at a constant rate of 100Hz to form data containing 27 channels. We exploit the sliding window technique and filter out 10 classes of data to obtain 17639 instances in total. In order to construct the problem situation in FedHealth 2, we use the Dirichlet distribution as in [Yurochkin *et al.*, 2019] to create disjoint non-iid. client

Client	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	avg
Base	**92.86**	17.68	**100.00**	**83.52**	18.78	77.66	**95.05**	17.58	**92.39**	**93.37**	29.12	**84.78**	**98.90**	24.18	**98.91**	**98.90**	41.44	**93.62**	**85.71**	37.02	69.07
FedAvg	60.27	62.36	50.56	73.98	74.27	62.90	64.03	87.78	74.49	64.71	65.24	63.35	68.33	64.79	63.12	85.26	66.21	59.64	67.87	72.46	67.58
FedBN	60.72	62.59	50.34	73.53	74.72	62.44	62.90	88.24	74.27	64.48	65.69	62.90	68.33	65.24	62.44	85.94	65.99	59.64	68.10	72.69	67.56
FedProx	60.50	62.36	50.34	73.98	73.81	61.76	63.57	87.78	74.27	64.71	66.37	63.12	68.33	65.69	62.44	85.49	66.21	59.41	67.87	72.46	67.52
FedPer	48.31	**97.51**	61.40	47.29	58.47	23.98	49.55	91.86	51.24	77.60	89.16	57.92	42.53	49.44	58.60	86.62	77.32	52.38	73.08	**97.52**	64.59
FedHealth 2	77.20	77.55	77.43	79.64	**81.94**	79.86	86.20	**95.02**	85.33	69.23	**91.42**	79.41	74.43	69.75	81.67	94.10	**82.77**	75.74	77.15	86.91	**81.14**
d-FedHealth 2	64.33	77.55	78.33	77.38	79.91	**80.77**	85.52	92.53	86.23	69.23	87.58	80.09	74.66	**70.43**	83.71	93.88	80.50	74.60	78.73	87.13	80.16
f-FedHealth 2	64.11	77.78	69.53	79.86	77.88	74.43	84.62	93.67	74.04	81.00	79.91	71.95	74.21	62.98	78.05	89.57	79.59	68.71	71.95	87.81	77.08

Table 1: Activity recognition results of 20 clients on PAMAP. Bold means the best result while underline means the second best result.

training data. Figure 5(a) visualizes how samples are distributed among 20 clients for PAMAP. In each client, half of the data are used to train and the remaining data are for testing.

COVID-19. We also adopt a public COVID-19 posterior-anterior chest radiography images dataset [Sait *et al.*, 2020]. This is a combined curated dataset of COVID-19 Chest X-ray images obtained by collating 15 public datasets and it contains 9,208 instances of four classes (1281 COVID-19 X-Rays, 3270 Normal X-Rays, 1656 viral-pneumonia X-Rays, and 3001 bacterial-pneumonia X-Rays) in total. In order to construct the problem situation in FedHealth 2, we split the dataset similar to PAMAP. Figure 5(b) visualizes how samples are distributed among 20 clients for COVID-19. In each client, half of the data are used to train and the remaining data are for testing.

4.2 Implementations Details and Comparison Methods

For PAMAP, we adopt a CNN for training and predicting. The network is composed of two convolutional layers, two pooling layers, two batch normalization layers, and two fully connected layers. For COVID-19, we adopt Alexnet [Krizhevsky *et al.*, 2012]. We use a three-layer fully connected neural network as the classifier with two BN layers after the first two fully connected layers following [Li *et al.*, 2021]. For model training, we use the cross-entropy loss and SGD optimizer with a learning rate of 10^{-2}. If not specified, our default setting for local update epochs is $E = 1$ where E means training epochs in one round. In addition, we randomly select 20% of the data to train a model of the same architecture as the pre-trained model.

We did not compare with FedHealth [Chen *et al.*, 2020] since our method focuses on more general setting where clients do not share large volume of datasets. We compare three extensions of our method with five methods including common federated learning methods and some federated learning methods designed for non-iid data particularly:

- Base: Each client uses local data to train local models.
- FedAvg [McMahan *et al.*, 2017]: The cloud aggregate all client models without any particular operations for non-iid data.
- FedBN [Li *et al.*, 2021]: Each client preserves the local batch normalization.
- FedProx [Li *et al.*, 2018]: Allow partial information aggregation and add a proximal term to FedAvg.
- FedPer [Arivazhagan *et al.*, 2019]: Each client preserves some local layers.

4.3 Classification Accuracy

The classification results for each client on PAMAP are shown in Table 1. From these results, we have the following observations: 1) No matter how the weights are calculated, our method achieves the best effects on average. It is obvious that our method significantly outperforms other methods with a remarkable improvement (over $\mathbf{10}\%$ on average). 2) In some clients, the base method achieves the best test accuracy. As it can be seen from Figure 5(a), the distributions on the clients are very inconsistent, which inevitably leads to generate the different difficulty levels in different clients. And some distributions in the corresponding clients are so easy that only utilizing the local data can achieve the ideal effects. 3) FedBN does not achieve the desired results. This could be caused by that FedBN is designed for the feature shifts while our experiments are mainly set in the label shifts.

Method	Base	FedAvg	FedBN	FedProx
avg	92.70	86.48	82.26	86.15
Method	**FedPer**	**FedHealth 2**	**d-FedHealth 2**	**f-FedHealth 2**
avg	88.51	94.82	**94.97**	93.5

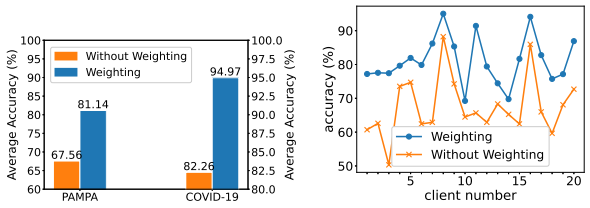
Table 2: Average accuracy of 20 clients on COVID-19

The classification results for each client on COVID-19 are shown in Table 2. From these results, we have the following observations: 1) No matter how the weights are calculated, our method achieves the best effects. However, in those experiments, our methods only have slight improvements. The classification accuracy only improves by 2% on average. It may be caused by that the COVID-19 dataset has much fewer classes compared with PAMAP. Henceforth, the task is much easier. 2) The base method achieves acceptable results in this dataset. From Figure 5(b), we can see that there are only a few classes in some clients, which means it is easy to obtain the desired results with few data. 3) FedBN gets the worst results. This demonstrates that FedBN is not good at dealing with label shifts. And FedBN does not consider the similarities among different clients.

4.4 Ablation Study

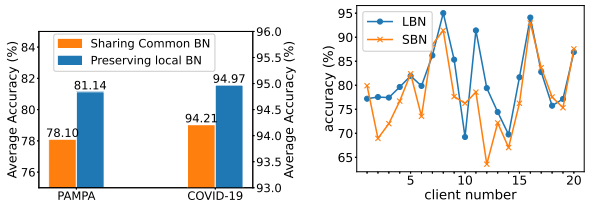
Effects of Weighting. To demonstrate the effect of weighting which considers the similarities among the different clients, we compare the average accuracy on PAMAP and COVID-19 between the experiments with it and without it. Without weighting, our method degenerates to FedBN. From

Figure 6(a), we can see that our method performs much better than FedBN which does not include the weighting part. Moreover, from Figure 6(b), we can see our method performs better than FedBN on all clients. These results demonstrate that our method with weighting can cope with the label shifts while FedBN cannot deal with this situation, which means our method is more applicable and effective.



(a) Average Accuracy (b) Client Acc on PAMAP

Figure 6: The effects of weighting.



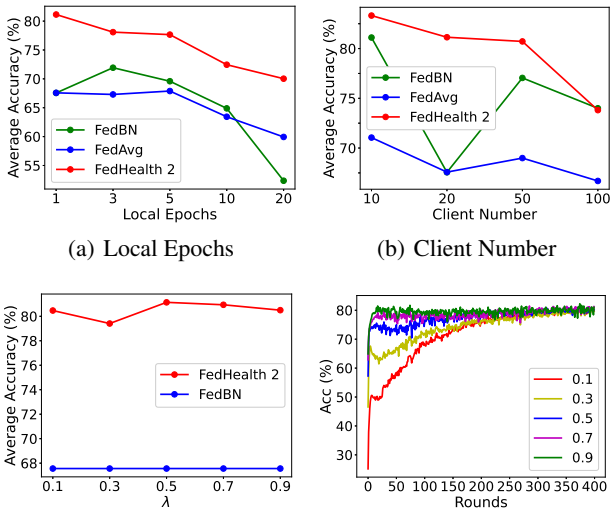
(a) Average Accuracy (b) Client Acc on PAMAP

Figure 7: The effects of preserving local batch normalization.

Effects of Preserving Local Batch Normalization. In this part, we illustrate the function of preserving local batch normalization. Figure 7(a) shows the average accuracy between the experiments with preserving local batch normalization and the experiments with sharing common batch normalization while Figure 7(b) shows the results on each client. LBN means preserving local batch normalization while SBN means sharing common batch normalization. Obviously, the improvements are not particularly significant compared with weighting. This may be caused by there mainly exist the label shifts in our experiments while preserving local batch normalization is for the feature shifts. However, our method still has a slight improvement, which shows the superiority of our method.

4.5 Parameter Sensitivity

In this section, we evaluate the parameter sensitivity of FedHealth 2 and we use FedHealth 2. Our method is affected by three parameters: local epochs, client number, and λ. We change one parameter and fix the other parameters. In Figure 8(a), we can see our method is best and it is descending with local epochs increasing, which may be caused that we keep the total number of the epochs unchanged and the communication among the clients are insufficient. From Figure 8(b), we can see that our method still achieves acceptable results. When the client numbers increase, our method goes down which may due to that few data in local clients make the weight estimation inaccurate. And we may take f-FedHealth 2 instead. Figure 8(c)-8(d) demonstrates λ slightly affects the



(a) Local Epochs (b) Client Number

(c) λ (d) λ

Figure 8: Influence of different hyper-parameters.

average accuracy of our method while it can change the convergence rate. The results reveal that FedHealth 2 is more effective and robust than other methods under different parameters in most cases.

4.6 Convergence

In this section, we investigate the convergence. In Figure 9, we can see our method almost convergences in the 10th round. And in the actual experiments, 20 rounds are enough for our method while FedBN needs over 400 rounds.

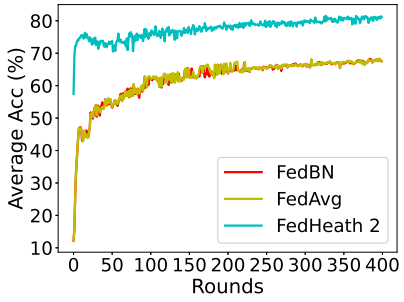


Figure 9: Convergence.

5 Conclusions And Future Work

In this article, we propose FedHealth 2, a weighted federated transfer learning algorithm via batch normalization for personalized healthcare. FedHealth 2 aggregates the data from different organizations without compromising privacy security and achieves relatively personalized model learning through combing considering similarities and preserving local batch normalization. Experiments have evaluated the effectiveness of FedHealth 2. FedHealth 2 follows FedHealth and continues to move forward in the direction of federated learning for healthcare. In the future, we plan to apply FedHealth 2 to more personalized and flexible healthcare.

References

[Arivazhagan *et al.*, 2019] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

[Chen *et al.*, 2017] Yiqiang Chen, Xiaodong Yang, Biao Chen, Chunyan Miao, and Hanchao Yu. Pdassist: Objective and quantified symptom assessment of parkinson's disease via smartphone. In *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 939–945. IEEE, 2017.

[Chen *et al.*, 2020] Yiqiang Chen, Xin Qin, Jindong Wang, Chaohui Yu, and Wen Gao. Fedhealth: A federated transfer learning framework for wearable healthcare. *IEEE Intelligent Systems*, 35(4):83–93, 2020.

[Dinh *et al.*, 2020] Canh T Dinh, Nguyen H Tran, and Tuan Dung Nguyen. Personalized federated learning with moreau envelopes. *arXiv preprint arXiv:2006.08848*, 2020.

[Inkster, 2018] Nigel Inkster. *China's cyber power*. Routledge, 2018.

[Ioffe and Szegedy, 2015] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[Krizhevsky *et al.*, 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[Li *et al.*, 2018] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018.

[Li *et al.*, 2019] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.

[Li *et al.*, 2021] Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. Fedbn: Federated learning on non-iid features via local batch normalization. *arXiv preprint arXiv:2102.07623*, 2021.

[McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

[Peyré *et al.*, 2019] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[Reiss and Stricker, 2012] Attila Reiss and Didier Stricker. Introducing a new benchmarked dataset for activity monitoring. In *2012 16th International Symposium on Wearable Computers*, pages 108–109. IEEE, 2012.

[Sait *et al.*, 2020] U. Sait, Lkv Gokul, S. Prajapati, R. Bhaumik, T Kumar, S Shivakumar, and K. Bhalla. Curated dataset for covid-19 posterior-anterior chest radiography images (x-rays). 2020.

[Segù *et al.*, 2020] Mattia Segù, Alessio Tonioni, and Federico Tombari. Batch normalization embeddings for deep domain generalization. *arXiv preprint arXiv:2011.12672*, 2020.

[Voigt and Von dem Bussche, 2017] Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10:3152676, 2017.

[Yang *et al.*, 2019] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[Yeganeh *et al.*, 2020] Yousef Yeganeh, Azade Farshad, Nassir Navab, and Shadi Albarqouni. Inverse distance aggregation for federated learning with non-iid data. In *Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning*, pages 150–159. Springer, 2020.

[Yu *et al.*, 2020] Tao Yu, Eugene Bagdasaryan, and Vitaly Shmatikov. Salvaging federated learning by local adaptation. *arXiv preprint arXiv:2002.04758*, 2020.

[Yurochkin *et al.*, 2019] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. In *International Conference on Machine Learning*, pages 7252–7261. PMLR, 2019.