MetaFed: Federated Learning Among Federations With Cyclic Knowledge Distillation for Personalized Healthcare

Yiqiang Chen<sup>1</sup>, Senior Member, IEEE, Wang Lu<sup>2</sup>, Xin Qin<sup>3</sup>, Jindong Wang<sup>4</sup>, and Xing Xie<sup>5</sup>, Fellow, IEEE

Abstract—Federated learning (FL) has attracted increasing attention to building models without accessing raw user data, especially in healthcare. In real applications, different federations can seldom work together due to possible reasons such as data heterogeneity and distrust/inexistence of the central server. In this article, we propose a novel framework called MetaFed to facilitate trustworthy FL between different federations. MetaFed obtains a personalized model for each federation without a central server via the proposed cyclic knowledge distillation. Specifically, MetaFed treats each federation as a meta distribution and aggregates knowledge of each federation in a cyclic manner. The training is split into two parts: common knowledge distillation and personalization. Comprehensive experiments on seven benchmarks demonstrate that MetaFed without a server achieves better accuracy compared with state-of-the-art methods [e.g., 10%+ accuracy improvement compared with the baseline for physical activity monitoring dataset (PAMAP2)] with fewer communication costs. More importantly, MetaFed shows remarkable performance in real-healthcare-related applications.

Index Terms—Federated learning (FL), healthcare, knowledge distillation (KD), personalization, transfer learning.

I. INTRODUCTION

MACHINE learning, especially deep learning, has become prominent in people’s daily lives [1], [2], [3]. It is applied to many health-related fields, such as human activity recognition [4], medical images [5], and other fields [6], [7], [8]. However, with the increasing awareness of data privacy and security, some countries and organizations released policies to prevent data leakage [9], [10]. In this situation, federated learning (FL) [11] was proposed and becomes increasingly popular.

Google [12] proposed the first FL algorithm called FedAvg to aggregate clients’ information. FedAvg replaces direct data exchanges with model parameter communication to preserve data privacy. Yang et al. [11] offer a review of FL and it groups FL into three categories, including horizontal FL, vertical FL, and transfer FL. Although FedAvg achieves promising performance in many applications, it may not be feasible in more challenging situations. For example, when meeting non-i.i.d. data, FedAvg will suffer from slow convergence and low accuracy. Therefore, some FL methods, e.g., FedBN [13], are proposed to solve these concrete problems.

In reality, the situations can be more difficult. Medical institutions may be grouped into multiple federations and no higher level governing organizations exist. Consider a real example. Several hospitals form a federation which is called the inner hospital federation while several medical companies, providing service to consumers at home, form another federation, which is called the outer hospital federation. Obviously, we can perform FedAvg inside both two federations. But, how can we combine these two federations? There is no higher level server and existing methods, e.g., FedAvg and FedBN, all fail in this situation. Fig. 1 gives an abstract summary of this real-challenging situation.

As shown in Fig. 1, a certain number of clients form a federation and different federations are independent enough that do not use a central server but communicate with each other instead. Inside each federation, different FL algorithms can be used to train a model. However, it remains unclear how to build personalized FL models outside the federations, i.e., FL among different federations. Moreover, data heterogeneity widely exists in these federations. We view each federation as a meta distribution and view the problem in this situation as meta FL.¹

In this article, we propose MetaFed, a meta FL framework for cross-federation FL. We focus on interfederation FL in this article and each federation can be viewed as an independent individual. To implement MetaFed, we propose a cyclic knowledge distillation (KD) method. MetaFed can solve data islanding and data statistical heterogeneity without requiring a server or sacrificing user privacy. Specifically, MetaFed consists of two stages, the common knowledge accumulation stage and the personalization stage. In the first stage, the model trained on the previous meta federation serves as the teacher for the next federation and KD [14], [15] aims to make use of the common information. Several rounds with fixed hyperparameters for KD are performed to ensure enough common

¹We use meta and federation interchangeably.

Manuscript received 23 November 2022; revised 20 March 2023; accepted 9 July 2023. Date of publication 28 July 2023; date of current version 30 October 2024. This work was supported in part by the National Key Research and Development Plan of China under Grant 2021YFC2501202, in part by the Natural Science Foundation of China under Grant 61972383 and Grant 62202455, in part by the Beijing Municipal Science and Technology Commission under Grant Z221100002722009, and in part by the Science Research Foundation of the Joint Laboratory Project on Digital Ophthalmology and Vision Science under Grant SYXK202201. (Corresponding author: Yiqiang Chen.)

Yiqiang Chen, Wang Lu, and Xin Qin are with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China (e-mail: yqchen@ict.ac.cn; lwu@ict.ac.cn).

Jindong Wang and Xing Xie are with the Microsoft Research Asia, Beijing 100080, China (e-mail: jindong.wang@microsoft.com; xingx@microsoft.com).

Digital Object Identifier 10.1109/TNNLS.2023.3297103
knowledge. In the personalization stage, we utilize KD with adapted hyperparameters to obtain personalized models for each federation. Through KD, it can not only acquire common knowledge among federations but also cope with feature shifts and label shifts. Moreover, MetaFed is extensible and can be deployed to many applications. The code for MetaFed is released at https://github.com/microsoft/PersonalizedFL.

Our contributions are as follows.

1) We propose MetaFed, a novel meta FL framework via cyclic KD for healthcare, which can accumulate common information from different federations without compromising privacy security, and achieve personalized models for each federation through adapted KD.

2) Comprehensive experiments on image and time-series datasets illustrate that MetaFed has remarkable performance in each federation without a server compared with state-of-the-art methods. Moreover, MetaFed reduces the number of rounds, thus saving communication costs.

3) Two real-health-related applications, Coronavirus disease 2019 (COVID-19) under label shifts and PD tremor under feature shifts, demonstrate that MetaFed can also achieve noteworthy performance in reality and MetaFed can be directly utilized in real-personalized FL.

4) MetaFed is extensible and can be applied in many healthcare applications, which means it can work well in many circumstances. We can even replace the KD with some other incremental learning methods for specific applications.

The remainder of this article is organized as follows. We introduce related work in Section II. In Section III, we elaborate on the proposed method and offer a clear summary. Then, the experimental implementations and results are presented to demonstrate the superiority of MetaFed in Section IV. Moreover, two real applications are provided in Section V to illustrate that our method can work well in reality. Finally, we conclude this article and provide some future directions in Section VI.

II. RELATED WORK

A. Machine Learning for Healthcare

Machine learning has shown its superior in many application fields, especially in healthcare [16], [17], [18]. From the simplest application where human daily activities are recognized [19] to the most difficult one where brain tumors can be classified [20], machine learning can exert its effectiveness. In a common framework, it often first collects lots of labeled data, then extracts features, and finally, trains a classifier. Unfortunately, the quality and number of collected labeled data in the first step often have a significant impact on the application. In many real applications, especially healthcare-related ones, obtaining centralized data is impossible. Although multiple persons or organizations generate a large amount of data, a few of them are willing to share their private data [21]. Moreover, a large number of regulations have emerged, such as [9] and [10], to protect data privacy and security. These factors make data form separate data islands where learning a model with aggregated data cannot work. In addition, heterogeneous data statistics can be another issue that seriously hinders the development of machine learning in healthcare [22]. Therefore, how to learn a robust model without directly exchanging data becomes a trend.

B. Federated Learning

To make full use of data in different separate clients and protect data privacy and security simultaneously, Google [12] first proposed FedAvg to train machine learning models via aggregating distributed mobile phones’ data with exchanging model parameters instead of directly exchanging data. FedAvg can work well with data islanding problems in many applications although it is simple. Subsequently, Yang et al. [11] wrote the first survey of FL research.

FL has attracted growing attention in many applications. And the traditional and simple FedAvg cannot satisfy complicated realistic scenes. When meeting data statistical heterogeneity, FedAvg may converge slowly and acquire large amounts of communication cost. Moreover, since only a shared global model is obtained, the model may degrade when predicting personalized clients. Some work tries to cope with these problems. FedProx [23] added a proximal term to FedAvg, which referred to the global model and allowed slight differences when training local models. Yu et al. [24] combined three traditional adaptation techniques: fine-tuning, multitask learning, and KD into federated models. Most recently, FedBN [13] tried to cope with feature shifts among clients by preserving local batch normalization parameters, which can represent data distributions to some extent. FED-ROD [25] paid attention to both generalization and personalization. Some other work made an effort to utilize personalization FL in the healthcare field [26], [27]. Chen et al. [26] proposed a federated transfer learning framework that needs some sharing data, while Lu et al. [27] proposed FedAP which could achieve personalized performance via aggregating with clients’ similarities. However, these methods need a server and have some limits in communication costs.

In this situation where no server exists, FedAvg even cannot be implemented. Sequential training may be a reasonable solution. Kopparapu and Lin [28] proposed FedFMC that dynamically forked devices into updating different global models and merged models in a lifelong way. Zacone et al. [29] leveraged the sequential training of subgroups of heterogeneous clients
to emulate the centralized paradigm. Zeng et al. [30] assigned clients to homogeneous groups to minimize the overall distribution divergence among groups. These methods still rely heavily on parallel FL, where sequential training round style with only one round and no closed loop is just an aid.

Some other work, e.g., [31], [32], [33], [34], communicated in a peer-to-peer environment without a server. BrainTorrent [31] presented a highly dynamic peer-to-peer environment, where all centers directly interacted with each other without depending on a central body. It seemed disorderly and chaotic, and it required lots of communication costs. Rieke et al. [32] considered key factors to FL, while FedH2L [33] utilized mutual distillation to exchange posteriors on a shared seed set between participants in a decentralized manner. Swarm learning [34] applied this style into the real decentralized and confidential clinical machine learning; however, these methods often require large communication costs, and a few are designed for personalization federation FL. No work pays attention to proposing a new paradigm for personalized FL among federations without a server.

C. Knowledge Distillation

KD has been a well-known technique to transfer knowledge since birth [14]. In the original version, the knowledge was transferred by mimicking the outputs of the teacher model on the same data. Later, besides imitating outputs, some work demonstrated that feature imitation could also guide the student model training [15]. A more detailed description of KD can be found in [35] and [36]. Nowadays, as a common technique, KD is also applied to FL [37], [38]. Though mimicking the global model and the local previous model, different implementations can be flexibly applied to different situations.

III. METHOD

A. Problem Formulation

In a personalized FL among federations setting, \( N \) different federations, denoted as \( \{F_1, F_2, \ldots, F_N\} \), have data, denoted as \( \{D_1, D_2, \ldots, D_N\} \), with different distributions, which means \( P(D_i) \neq P(D_j) \). For simplicity, we only study the case where the input and output spaces are the same, i.e., \( \mathcal{X}_i = \mathcal{X}_j, \mathcal{Y}_i = \mathcal{Y}_j, \forall i \neq j \). Each dataset, \( D_i = \{(x_{ij}, y_{ij})\}_{j=1}^{n_{train}} \), consists of three parts, a train dataset \( D_i^{train} = \{(x_{ij}^{train}, y_{ij}^{train})\}_{j=1}^{n_{train}} \), a validation dataset \( D_i^{valid} = \{(x_{ij}^{valid}, y_{ij}^{valid})\}_{j=1}^{n_{valid}} \) and a test dataset \( D_i^{test} = \{(x_{ij}^{test}, y_{ij}^{test})\}_{j=1}^{n_{test}} \). We have \( n_i = n_{train}^{i} + n_{valid}^{i} + n_{test}^{i} \) and \( D_i = D_i^{train} \cup D_i^{valid} \cup D_i^{test} \).

We aim to combine information of all federations without data exchange to learn a good model \( f_i \) for each federation on its local dataset \( D_i \).

\[
\min_{\{f_i\}_{i=1}^N} \frac{1}{N} \sum_{i=1}^{N} \frac{1}{n_i} \sum_{j=1}^{n_{test}} \ell\left(f_i(x_{ij}), y_{ij}^{test}\right) \tag{1}
\]

where \( \ell \) is a loss function.

B. Overview of MetaFed

Consider the union of different federations where there is no server among them and distribution shifts exist. How to make them communicate equally without any other governors and share common knowledge without direct data exchange is the key. MetaFed aims to accumulate common knowledge and preserve personalized information without compromising data privacy and security via KD in a cyclic way. Fig. 2 gives an overview.

Without loss of generality, we assume there are four federations, and it can be extended to the more general case easily. As shown in Fig. 2, the whole training process is split into two stages, the common knowledge accumulation stage (blue arrows) and the personalization stage (green arrows). In the common knowledge accumulation stage, the federations are trained in order and the previously trained one serves as the teacher for the next one. The common knowledge accumulation stage lasts for several rounds to ensure each federation’s common knowledge are extracted completely. The personalization stage is also trained in the same style but the model is sent to the next federation without local training for losing no common knowledge. From Fig. 2, we can see no server participates in the training process. The two stages are both based on feature KD (as shown in Fig. 3)

\[
\ell_{dist}(g_{tea}, g_{stu}; x) = \|g_{tea}(x) - g_{stu}(x)\|_2^2 \tag{2}
\]

where \( g_{tea} \) is the feature extractor of the previous federation while \( g_{stu} \) is for the current training federation, and \( x \) is a sample of data from the current federation. Through KD, we can make good use of knowledge, viewed as common knowledge, from the previous federation. Therefore, the total loss to train the local model, \( f_i \), is

\[
\ell_{total}^i = \frac{1}{n_{train}^{i}} \sum_{(x, y) \in D_i^{train}} \ell_{cls}(f_i; x, y) + \lambda \ell_{dist}(g_{tea}, g_i; x) \tag{3}
\]

where \( \lambda \) is a tradeoff of knowledge transfer and focusing on the current data, while \( \ell_{cls} \) is the cross-entropy loss. \( f_i = c_i \circ g_i \), where \( c_i \) is the classification layer and \( g_i \) is the feature extractor. In the following, we will specify the two stages, respectively, and illustrate how to design \( \lambda \) in detail.
D. Personalization Stage

This stage happens in the second part of the whole training process. In the above-mentioned stage, we obtain the common model $f$, which contains enough common knowledge. Since no server exists, we have to obtain the personalization models in the same style (sequential) as the first stage. To prevent common knowledge loss, we transmit the common $f$ to the next federation before local training. The specific detail of the second stage can be found in Algorithm 2.

When the common model performs seriously terribly on the validation data of the current federation, we want to refer little to it and thereby set $\lambda = 0$. In the first stage, the current $f_i$ has contained other federations’ knowledge. When the common model’s performance is acceptable on the current validation data, we adapt $\lambda$ for personalization

$$\lambda = \lambda_0 \times 10^{\min(1,(acc_{\text{valid}_{i+1}}-acc_{\text{valid}_{i+1}})5)-1}. \tag{4}$$

Compared with the local model’s performance, the better the common model’s performance is, the larger $\lambda$ will be.

E. Summary

We propose a novel framework called MetaFed to facilitate trustworthy FL between different federations. Here, we re-emphasize these specially designed techniques.

a) How to ensure enough common knowledge accumulation in clients, the following techniques can be adopted:

1) Perform common knowledge accumulation with several rounds.
2) Transmit enough teacher information according to copy or distillation with fixed $\lambda$.
3) Incorporate local information.

b) How to discard redundant information and obtain personalized models:

1) Dynamically adjust the amount of inherited information in personalization.
2) Reduce the influence of other federations’ information.

c) How to determine inheritance or discard:

1) Valid accuracy of each federation can serve as guidance. With these flexible techniques, our framework can ensure personalization without a central server.

F. Extensibility

Surprisingly, MetaFed can be a more flexible framework, just as shown in Fig. 4, and we call this universal situation as MetaFed++. MetaFed++ imitates the Internet to transmit information. Federations are scattered around and...
possible connecting paths are used to transmit information. MetaFed++ categorizes different federations into multiple groups and performs both intergroup and intragroup communications via cyclic KD. The graph structure is flexible and changeable. We can even treat groups as new federations and perform MetaFed++ among groups. Besides, link paths can also be dynamic. In reality, we can group by geography, organizations, and some other real rules. In experiments, we can group by similarities among feature statistics following FedAP [27]. Obviously, we can easily extend MetaFed to MetaFed++ for larger and more flexible real-world applications.

G. Time Complexity

Denote the number of communication rounds, local train iterations per round, time costs per iteration, and federations as $R_I$, $L_I$, $T_I$, and $N$, respectively. The time complexity for our method can be $O(R_I \times L_I \times T_I \times N)$ similar to FedAvg. However, compared with FedAvg, with limited communication rounds, our method can work better.

IV. EXPERIMENTS

We evaluate the performance of MetaFed on seven benchmarks, including time series and image modalities. The first three benchmarks are on feature shifts where distribution shifts exist in the input space while the other four benchmarks are on label shifts (i.e., distribution shifts exist in the output space).

We compare our method with three state-of-the-art methods, including common FL methods and some FL methods designed for non-i.i.d. data.

1) FedAvg [12]: Directly aggregate models’ parameters without personalization.
2) FedProx [23]: Allow slight differences between the local model and the global model via a proximal term added to FedAvg.
3) FedBN [13]: Preserve the local batch normalization not affected by the other clients.

Since these methods all need a server, we ease this restriction for them. Adapting these methods without a server will increase communication costs with no performance improvement. All methods use the same model for fairness.

A. Experiments on Feature Shifts

1) VLCS:

![Image of graph structure]

2) PACS:

a) Dataset: PACS [41] is another popular object classification benchmark. It also consists of four subdatasets (photographs, art painting, cartoon, and sketch) with 9991 images of seven classes. There exist large discrepancies in image styles among different subdatasets. Similar to VLCS, we choose 10% for training, 10% for validation, and 20% for testing. We also utilize validation parts for the guidance and selections.

b) Implementation details: We utilize the same architecture and the same optimizer as VLCS. All methods are implemented in the same environment for fairness and we run three trials to record the average accuracy.

c) Results: The classification results for each federation on PACS are shown in Table II. We have the following observations from these results: 1) our method achieves the best performance in each federation which demonstrates the superiority and the capability of personalization of our method. Due to the inherent inconsistency of federation data, certain methods may perform better on certain federations, e.g., FedProx on VOC2007. 2) Since it is a feature shift situation, FedBN has a better performance compared with FedAvg. FedProx has an acceptable performance.

2) VOC2007:

a) Dataset: VOC2007 [38] is a popular public object detection dataset. It comprises 10,729 images of five classes. Each subdataset serves as one federation and there are four federations in total. Since each dataset contains too many images, we choose 10% for training, 10% for validation, and 20% for testing. The validation parts are utilized to guide training and select the best model for each federation.

b) Implementation details: For VOC2007, we adopt AlexNet [40] as the feature extractor and a three-layer fully connected neural network as the classifier. For model training, we use a stochastic gradient descent (SGD) optimizer with a learning rate of $10^{-3}$. For our method, we search best hyperparameters within $[0.1, 1, 5, 10]$ for $\lambda_0$ and $[0.4, 0.5, 0.6, 1.1]$ for $l_1$. More analysis on hyperparameter sensitivity can be found in Section IV-C2. All methods are implemented in the same environment for fairness and we run three trials to record the average accuracy.

c) Results: The classification results for each federation on VOC2007 are shown in Table I. We have the following observations from these results: 1) without a doubt, our method achieves the best performance in each federation (over 22% compared with FedBN). We even almost achieve the best performance in each federation which demonstrates the superiority and the capability of personalization of our method. Due to the inherent inconsistency of federation data, certain methods may perform better on certain federations, e.g., FedProx on VOC2007. 2) Since it is a feature shift situation, FedBN has a better performance compared with FedAvg. FedProx has an acceptable performance.

TABLE I

| Method   | Caltech101 | LabelMe | SUN09 | VOC2007 | AVG    |
|----------|------------|---------|-------|---------|--------|
| FedAvg   | 82.69      | 54.43   | 51.52 | 44.89   | 58.38  |
| FedProx  | 83.04      | 55.74   | 51.98 | 47.70   | 59.62  |
| FedBN    | 90.81      | 54.80   | 50.15 | 44.30   | 60.02  |
| MetaFed  | 94.35      | 59.32   | 57.62 | 45.78   | 64.27  |

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
FedProx. Different from VLCS, FedAvg and FedProx are far worse than FedBN, which can be due to larger divergences, more classes, and so on.

3) Human Activity Recognition:

a) Dataset: To further prove the superiority of our method, we also adopt a public time-series benchmark with feature shifts, physical activity monitoring dataset (PAMAP2) [42]. PAMAP2 is human activity recognition with data from 18 human activities performed by nine subjects. We use three inertial measurement units’ data with 27 channels and utilize the sliding window technique to preprocess data. Twelve classes with 12,291 instances are selected. Since there are no natural subdatasets in PAMAP2, we split artificially PAMAP2 into four subdatasets according to persons and we call this task a cross-person task [43]. Nine subjects are divided into four groups, [3, 2, 8], [1, 5], [0, 7], and [4, 6], where different numbers denote different persons, and we try our best to make each domain have a similar number of data. Similar to VLCS, we choose 10% for training, 10% for validation, and 20% for testing. We also utilize validation parts for the guidance and selections.

b) Implementation details: For PAMAP2, we utilize a CNN composed of two convolutional layers, two pooling layers, two batch normalization layers, and two fully connected layers [19]. Other settings are similar to VLCS.

c) Results: The classification results for each federation on PAMAP2 in the cross-person setting are shown in Table III. We have the following observations from these results: 1) our method achieves the best performance on average with an improvement of 1.04%. However, it is slightly worse than FedBN in Federation 0 and slightly worse than FedProx in Federation 3. We think that the task is so simple that some methods, e.g., FedProx, with few techniques for feature shifts can work slightly better. And the improvements on some federations with a drop of less than 1% do not illustrate that our method is unacceptable. 2) Similar to VLCS, in this feature shift situation, FedBN has a better performance compared FedAvg and FedProx. Simultaneously, FedAvg and FedProx have a similar performance to FedBN.

4) Summary: In the feature shift situation, whatever data are visual images or time series, our method achieves the best performance compared with other methods. The more difficult the task is, the larger improvements our method has. Moreover, FedBN, a method specifically designed for the feature shift, has the second-best performance.

B. Experiments on Label Shifts

1) Human Activity Recognition:

a) Dataset: For the label shift benchmarks, we first adopt a public time-series benchmark, PAMAP2 [42]. PAMAP2 contains data on 18 human activities performed by nine subjects. We also use three inertial measurement units’ data with 27 channels and utilize the sliding window technique to preprocess data. Ten classes with 17,639 instances are selected. To simulate label shift, we follow [44] and use Dirichlet distributions to create disjoint non-i.i.d. training data. Fig. 5(a) visualized how samples are distributed. For each federation, 40%, 30%, and 30% of data are used for training, validation, and testing, respectively.

b) Implementation details: For PAMAP2, we utilize a CNN composed of two convolutional layers, two pooling layers, two batch normalization layers, and two fully connected layers [19]. Other settings are similar to VLCS.

c) Results: The classification results for each federation on PAMAP2 are shown in Table IV. We have the following observations from these results: 1) our method also achieves the best effects on average with a remarkable improvement (over 2.63% compared with FedBN) in this situation where label shifts exist. 2) In this situation, 20 federations in total make the problem more complicated. Although our method does not achieve the best performance in each federation, it still achieves acceptable results in almost every federation. 3) Compared with FedAvg and FedProx, FedBN and our method...
TABLE V

ACCURACY ON THREE BENCHMARKS OF MedMNIST. BOLD MEANS THE BEST RESULT

| Dataset | OrganA | OrganC | OrganS |
|---------|--------|--------|--------|
| Client  | FedAvg | FedBN  | MetaFed | FedAvg | FedBN  | MetaFed | FedAvg | FedBN  | MetaFed |
| 0       | 94.20  | 93.75  | 94.09   | 96.02  | 86.40  | 87.25   | 82.44  | 90.65  | 52.39  | 52.13   | 75.00  | 96.01  |
| 1       | 91.23  | 91.34  | 92.03   | 95.79  | 90.00  | 90.29   | 94.86  | 96.29  | 50.26  | 51.59   | 76.98  | 98.41  |
| 2       | 92.61  | 92.95  | 93.40   | 97.72  | 88.03  | 85.76   | 92.02  | 93.16  | 67.47  | 69.07   | 75.20  | 82.40  |
| 3       | 89.85  | 90.76  | 90.31   | 95.55  | 84.39  | 87.82   | 91.50  | 94.05  | 67.82  | 67.82   | 79.79  | 88.83  |
| 4       | 96.25  | 96.47  | 97.27   | 97.95  | 84.05  | 85.47   | 86.32  | 88.89  | 75.67  | 77.01   | 79.95  | 85.83  |
| 5       | 88.62  | 89.65  | 88.96   | 92.61  | 90.31  | 89.74   | 92.88  | 90.60  | 53.33  | 53.33   | 54.13  | 89.87  |
| 6       | 90.44  | 90.79  | 92.49   | 96.02  | 78.35  | 79.20   | 78.92  | 82.62  | 82.98  | 81.38   | 86.97  | 96.28  |
| 7       | 94.65  | 94.65  | 95.90   | 96.70  | 95.16  | 96.58   | 97.44  | 97.72  | 61.97  | 59.84   | 67.29  | 98.67  |
| 8       | 92.84  | 93.30  | 94.89   | 98.18  | 83.52  | 84.09   | 86.36  | 90.34  | 82.35  | 82.89   | 92.51  | 94.92  |
| 9       | 89.29  | 89.29  | 89.18   | 94.31  | 91.45  | 90.88   | 90.03  | 95.73  | 90.72  | 90.19   | 99.20  | 99.73  |
| 10      | 91.81  | 91.81  | 92.15   | 96.93  | 85.76  | 86.89   | 85.19  | 86.89  | 65.78  | 59.36   | 75.67  | 88.50  |
| 11      | 92.94  | 91.69  | 91.69   | 93.96  | 90.03  | 91.74   | 90.60  | 94.02  | 83.42  | 86.10   | 86.90  | 90.64  |
| 12      | 89.62  | 90.99  | 91.11   | 95.33  | 86.04  | 87.18   | 83.76  | 89.17  | 73.53  | 71.12   | 76.47  | 84.22  |
| 13      | 93.84  | 93.50  | 92.25   | 96.69  | 90.63  | 91.76   | 92.90  | 95.74  | 90.13  | 89.33   | 90.93  | 97.07  |
| 14      | 93.86  | 92.83  | 92.72   | 97.50  | 86.36  | 87.50   | 85.80  | 91.76  | 83.11  | 83.65   | 89.28  | 89.81  |
| 15      | 89.65  | 89.65  | 90.33   | 94.99  | 91.43  | 92.29   | 94.29  | 93.71  | 46.42  | 45.89   | 72.15  | 81.96  |
| 16      | 93.52  | 92.83  | 93.52   | 97.38  | 86.89  | 87.75   | 88.03  | 93.16  | 44.80  | 43.20   | 57.87  | 84.00  |
| 17      | 90.43  | 90.43  | 91.00   | 94.99  | 92.33  | 90.91   | 92.90  | 95.74  | 61.17  | 62.77   | 69.95  | 88.03  |
| 18      | 92.26  | 92.04  | 92.04   | 97.16  | 85.76  | 86.04   | 82.34  | 87.18  | 56.38  | 58.78   | 73.67  | 96.28  |
| 19      | 92.38  | 91.92  | 91.13   | 97.27  | 91.76  | 92.33   | 92.90  | 94.03  | 79.41  | 76.68   | 87.43  | 94.12  |
| AVG     | 92.01  | 92.03  | 92.32   | 96.15  | 88.13  | 88.57   | 89.07  | 92.07  | 68.46  | 68.26   | 78.37  | 91.28  |

Fig. 5. Number of samples per class allocated to each federation (indicated by dot size). (a) PAMAP2. (b) OrganA. (c) OrganC. (d) OrganS.

Fig. 6. Ablation study on PAMAP2 under feature shifts. (a) Average Acc. (b) Federation Acc.

Fig. 7. Ablation study on PAMAP2 under label shifts. (a) Average Acc. (b) Federation Acc.

achieve remarkable improvement, which may illustrate that FedBN can also cope with label shifts sometimes.

2) Medical Image Classification:

a) Dataset: To further validate our method, we evaluate our method on three public medical image classification benchmarks. We choose three datasets, OrganAMNIST, OrganCMNIST, and OrganSMNIST [45], [46], from a larger scale MNIST-like collection of standardized biomedical images, MedMNIST [47], [48]. These three datasets are all about Abdominal CT images with 11 classes and they have 58,850, 23,660, and 25,221 samples, respectively. Similar to PAMAP2, we utilize the Dirichlet distribution to split data, and Fig. 5(b)–(d) visualizes how samples are distributed. In each federation, 40%, 30%, and 30% of data are used for training, validation, and testing, respectively.

b) Implementation details: For these three datasets, we utilize adapted LeNet5 [49] due to the image size of 28 × 28. Other settings are similar to VLCS.

c) Results: The classification results for each federation on OrganAMNIST, OrganCMNIST, and OrganSMNIST are shown in Table V. We have the following observations from these results: 1) our method also achieves the best effects on average with a remarkable improvement (over 3.83%, 3.00%, and 12.91%, respectively) in this situation where label shifts exist. 2) When federation distributions have small differences [Fig. 5(b) and (c)], three state-of-the-art meth-
ods have similar performance and ours achieves remarkable improvements. When federations have huge differences from each other [Fig. 5(d)], FedBN can achieve a remarkable improvement compared with FedAvg and FedProx, while ours shows another crazy improvement compared with FedBN. The above-mentioned experiments demonstrate that our method can achieve the best performance in both settings.

3) Summary: In the label shift situation, whatever data are visual images or time series, our method also achieves the best performance compared with other methods. Interestingly, FedBN, a method specifically designed for the feature shift, still has the second-best performance.

C. Analysis

1) Ablation Study: We also perform an ablation study to illustrate the effects of each part of our methods. As shown in Figs. 6 and 7, we can see that replacing KD with fine-tuning (fine-tune), training without common knowledge accumulation stage (W.O. I), and training without personalization stage (W.O. II) will all make performance drop, which demonstrates that each part of our method can all bring benefits. We do not use the common f for testing but each local model with local adaptation brought by KD does testing. There also exist slight differences between results under feature shifts and label shifts. When feature shifts exist, the common knowledge accumulation stage may play a more important role compared with the personalization stage and there is a further improvement with the personalization stage. Fine-tune seems to bring no improvement. When label shifts exist, fine-tune can bring slight improvements while both two stages work well.

2) Parameter Sensitivity: In this part, we evaluate five hyperparameters, \( \lambda_0 \), \( l_1 \), feature distillation styles,\(^3\) whether to share BN parameters, and the transmission sorts. We change one parameter and fix the others. From Fig. 8(a)–(d), we can see that our method can achieve better performance than FedAvg under label shifts whatever each hyperparameter is. Among five hyperparameters, whether to share BN parameters has minimal impact on results. We also perform parameter sensitivity on PAMAP2 under feature shifts. As shown in Fig. 8(f), our method still achieves better performance compared with FedAvg under feature shifts. Moreover, similar to the results under label shifts, whether to share BN parameters has a slight impact on the final results. The results reveal that MetaFed is more effective and robust than other methods under different hyperparameters in most cases.

3) Communication Costs: To further prove that our method can reduce communication costs, we increase the local training iteration number and decrease the total communication rounds to evaluate our method and the baseline. As shown in Fig. 9(a), when communication costs are limited under feature shifts, our method has related stable results, while FedAvg drops seriously. A similar phenomenon exists in the situation under label shifts, as shown in Fig. 9(b). In realistic applications, the communication cost is an important evaluation metric and there are often strictly limited costs. Therefore, few communication costs with stable and acceptable performance are vital.

4) MetaFed++: We evaluate MetaFed++ in this part. We group 20 federations into three groups and perform MetaFed in both intergroup and intragroup. As shown in Fig. 9(c) and (d), MetaFed++ even achieves a better average accuracy compared with MetaFed, which means our method can be easily deployed in real-large applications. In addition, although MetaFed++ performs worse than MetaFed in some federations, MetaFed++ performs better than MetaFed in most federations. This can be due to that federations are more likely to be influenced by the federations within the same groups.

5) Convergence: We offer the convergence analysis in this part. As shown in Fig. 9(e), both the average loss and the average accuracy are convergent and they end in stable and ideal states, which demonstrates the superiority of MetaFed.

6) Compare With More Methods: In this part, we provide some comparisons with BrainTorrent [31], a peer-to-peer
TABLE VI
Comparison With BrainTorrent on PAMAP2

|                | BrainTorrent | MetaFed |
|----------------|--------------|---------|
| Feature shifts | 80.59        | 86.07   |
| Label shifts   | 83.55        | 90.07   |

Fig. 10. Chest X-ray images. (a) Normal. (b) Pneumonia-Bacterial. (c) Pneumonia-Viral. (d) COVID-19.

method. As shown in Table VI, our method still achieves better performance under both feature shifts and label shifts. Besides, our method saves storage space compared with BrainTorrent.

V. Real-World Applications

A. COVID-19

1) Background: COVID-19 is a contagious disease caused by a virus [50]. According to the Oxford English Dictionary [51], it is mainly characterized by fever and cough. It is capable of progressing to pneumonia, respiratory and renal failure, blood coagulation abnormalities, and death, especially in the elderly and people with underlying health conditions. The first known case was identified in Wuhan, China, in December 2019, and then the disease quickly spread worldwide, resulting in the COVID-19 pandemic [50]. There exist several ways to diagnose COVID-19, including viral testing, imaging, coding, pathology, and so on. Chest CT scans may be one of the most useful ways to detect it [52]. Fig. 10(a)–(d) gives an example of chest X-ray images.

2) Experiment:

a) Dataset: COVID-19 is a public COVID-19 posterior–anterior chest radiography image dataset [53]. It is a combined curated chest X-ray image dataset that collates 15 public datasets. Four classes (1281 COVID-19 X-rays, 3270 normal X-rays, 1656 viral-pneumonia X-rays, and 3001 bacterial-pneumonia X-rays) exist in the dataset and there are 9208 images in total. We utilize this public benchmark to evaluate our method with a real health-related application under label shifts. Similar to medical image classification, we utilize the Dirichlet distribution to split data, and Fig. 11 visualizes how samples are distributed. In each federation, 40%, 30%, and 30% of data are used for training, validation, and testing, respectively.

Fig. 11. Number of samples per class allocated to each federation (indicated by dot size) on COVID-19.

TABLE VII
Accuracy on COVID-19. Bold means the best results

| Client | FedAvg | FedProx | FedBN | MetaFed |
|--------|--------|---------|-------|---------|
| 0      | 74.45  | 76.64   | 70.07 | 78.10   |
| 1      | 95.59  | 97.06   | 92.65 | 99.26   |
| 2      | 87.50  | 85.29   | 91.18 | 91.91   |
| 3      | 89.05  | 90.51   | 93.43 | 97.81   |
| 4      | 88.32  | 89.05   | 95.62 | 97.08   |
| 5      | 84.56  | 86.76   | 85.29 | 97.06   |
| 6      | 97.81  | 98.54   | 93.43 | 97.81   |
| 7      | 73.53  | 75.74   | 69.12 | 80.88   |
| 8      | 86.86  | 87.59   | 91.97 | 91.97   |
| 9      | 87.50  | 89.71   | 94.12 | 100.00  |
| 10     | 83.94  | 83.94   | 87.59 | 88.32   |
| 11     | 81.02  | 84.67   | 83.21 | 94.16   |
| 12     | 97.79  | 97.06   | 93.38 | 97.79   |
| 13     | 73.72  | 73.72   | 55.47 | 71.53   |
| 14     | 85.40  | 86.86   | 91.24 | 95.62   |
| 15     | 84.56  | 88.97   | 91.18 | 97.79   |
| 16     | 93.43  | 94.16   | 89.78 | 96.35   |
| 17     | 86.13  | 87.59   | 89.78 | 89.78   |
| 18     | 96.38  | 96.38   | 91.30 | 98.55   |
| 19     | 75.91  | 75.91   | 64.96 | 78.10   |
| AVG    | 86.17  | 87.31   | 85.74 | 91.99   |

b) Implementation details: For this real application, we utilized AlexNet, and other settings are similar to VLCS.

c) Results: The classification results for each federation on COVID-19 are shown in Table VII. We have the following observations from these results: 1) our method achieves the best performance in this real health-related application. Compared with the second-best method, FedProx, MetaFed has a significant improvement of 4.68%. Moreover, our method almost achieves the best performance in every federation. 2) In this real application under label shifts, FedBN does not achieve the second-best performance, since it is originally designed for feature shifts. Instead, FedProx achieves the second best, while FedAvg even performs better than FedBN.

B. Parkinson Disease

1) Background: Parkinson’s disease (PD) is a long-term degenerative disorder of the central nervous system [54]. It leads to a significant drop in the patient’s quality of life and has drawn increasing attention for the substantial morbidity, increased mortality, and high economic burden [55]. Until now, there exist no disease-modifying drugs (drugs that target the causes or damage) are approved for Parkinson’s. Large
amounts of costs are paid for PD, as a study based on 2017 data estimated that the U.S. economic PD burden at $51.9 billion, including direct medical costs of $25.4 billion and $26.5 billion in indirect and nonmedical costs [56]. According to [57], patients with PD often suffer from four major motor symptoms, including tremors, rigidity, bradykinesia, and postural instability.

2) **PDAssist** To reduce the pressure on doctors, our team constructed a mobile PD assessment tool, PDAssist [58]. PDAssist is based on the unified PD rating scale (UPDRS) [59] and it can assess PD symptoms automatically. As shown in Fig. 12(a), the patient does some specially designed tasks. At the same time, the mobile phone collects corresponding data and uploads them to the hospital where the patient belongs to. The cloud in the corresponding hospital will offer an estimation score, from 0 to 4 (from normal to severe). Then, the corresponding doctor can directly have an estimation of the patient’s situation. This tool can greatly reduce the pressure on doctors and brings patients a better medical experience.

3) **Experiment:**

a) **Dataset:** In this article, we mainly focus on the tremor, a typical symptom of PD. We collect data via PDAssist and the corresponding tasks are shown in Fig. 12(b). We totally collect 11,616 time-series samples from 167 patients. Each sample contains six channels of two sensors, an accelerometer and a gyroscope, and the dimension is $6 \times 32$. Since 167 patients are distributed in 16 hospitals and each hospital cannot directly exchange data with others, an FL method is required to construct personalized models for each hospital. More notably, due to different hobbies, lifestyles, body shapes, devices, and so on, there naturally exist feature shifts in data among different hospitals. Simplified, in this task, we only predict whether the person is with PD.

b) **Implementation details:** For this real application, we utilize a CNN composed of two convolutional layers, two pooling layers, two batch normalization layers, and two fully connected layers. Please note the kernel sizes are different between this task and PAMAP2 and other settings are similar to VLCS.

c) **Results:** The classification results for each federation on tremor of PD are shown in Table VIII. We have the following observations from these results: 1) our method still achieves the best performance in this real application. And it has improvements of 16.11%, 16.10%, and 6.48% compared with FedAvg, FedProx, and FedBN, respectively. 2) In this real application under feature shifts, FedBN achieves the second-best performance, while FedAvg and FedProx perform terribly.

C. **Summary**

In these two real-health-related applications, our method achieves the best performance compared with other state-of-the-art methods, which demonstrates the superiority of our method. FedBN performs worst under label shifts different from Section IV, while it still achieves the second best under feature shifts.

VI. **Conclusion and Future Work**

In this article, we proposed MetaFed which uses cyclic KD for meta FL. MetaFed organizes federations in another novel style that does not require a central server. Comprehensive experiments have demonstrated the effectiveness of MetaFed. Moreover, MetaFed has shown its superiority in real-health-related applications. In the future, we plan to combine MetaFed with common methods, such as FedAvg, to implement a complete FL system, including intrafederation and interfederation. We also plan to apply MetaFed for heterogeneity architectures and more realistic healthcare applications.

**REFERENCES**

[1] W. Lu, J. Wang, and Y. Chen, “Local and global alignments for generalizable sensor-based human activity recognition,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), May 2022, pp. 3833–3837.

[2] W. Lu, J. Wang, Y. Chen, S. J. Pan, C. Hu, and X. Qin, “Semantic-discriminative mixup for generalizable sensor-based cross-domain activity recognition,” Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol., vol. 6, no. 2, pp. 1–19, Jul. 2022.

[3] M. He, J. Zhang, S. Shan, X. Liu, Z. Wu, and X. Chen, “Locality-aware channel-wise dropout for occluded face recognition,” IEEE Trans. Image Process., vol. 31, pp. 788–798, 2022.

[4] W. Lu, Y. Chen, J. Wang, and X. Qin, “Cross-domain activity recognition via substructural optimal transport,” Neurocomputing, vol. 454, pp. 65–75, Sep. 2021.
[56] W. Yang et al., “Current and projected future economic burden of Parkinson’s disease in the U.S.,” NPJ Parkinson’s Disease, vol. 6, no. 1, pp. 1–9, 2020.
[57] J. Jankovic, “Parkinson’s disease: Clinical features and diagnosis,” J. Neurol., Neurosurg. Psychiatry, vol. 79, no. 4, pp. 368–376, 2008.
[58] Y. Chen, X. Yang, B. Chen, C. Miao, and H. Yu, “PsyAssist: Objective and quantified symptom assessment of Parkinson’s disease via smartphone,” in Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM), Nov. 2017, pp. 939–945.
[59] C. G. Goetz, “The unified Parkinson’s disease rating scale (UPDRS): Status and recommendations,” Movement Disorders, vol. 18, no. 7, pp. 738–750, Jul. 2003.

Yiqiang Chen (Senior Member, IEEE) received the B.S. and M.S. degrees in computer science from Xiangtan University, Xiangtan, China, in 1996 and 1999, respectively, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences (ICT, CAS), Beijing, China, in 2003. In 2004, he was a Visiting Scholar Researcher with the Department of Computer Science, The Hong Kong University of Science and Technology, Hong Kong. He is currently a Professor with ICT, CAS, where he is also the Director of the Research Center for Ubiquitous Computing Systems. He has authored or coauthored over 200 publications in several top journals and conferences, including IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (TKDE), IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, VOL. 35, NO. 11, NOVEMBER 2024

Wang Lu received the B.S. degree in mathematics and applied mathematics from Tsinghua University, Beijing, China, in 2016. He is currently pursuing the Ph.D. degree with the Research Center for Ubiquitous Computing Systems, Institute of Computing Technology, Chinese Academy of Sciences, Beijing. His research interests include transfer learning, federated learning, and activity recognition.

Xin Qin received the B.S. and M.S. degrees in computer science and technology from Shandong Normal University, Jinan, China, in 2015 and 2018, respectively. She is currently pursuing the Ph.D. degree with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. Her current research interests include machine learning, pervasive computing, and activity recognition.

Jindong Wang received the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2019. He was a Visiting Student with The Hong Kong University of Science and Technology, Hong Kong, in 2018. He is currently a Senior Researcher with Microsoft Research Asia, Beijing. His research interests include transfer learning, machine learning, data mining, and ubiquitous computing.

Xing Xie (Fellow, IEEE) is currently a Senior Principal Research Manager with the Microsoft Research Asia, Beijing, China. He has authored or coauthored over 300 referred journals and conference papers during the past years. His research interests include data mining, social computing, and ubiquitous computing.

Dr. Xie is also a Distinguished Member of ACM and a fellow of the China Computer Federation (CCF). He has received the Best Paper Award from Ubiquitous Intelligence and Computing (UIC) 2010 and ICDM 2013, the Best Student Paper Award from ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD) 2016, and the Ten-Year Impact Award from the ACM SIGSPATIAL 2019. He served as the Program Co-Chair of ACM Ubicomp 2011, UIC 2015, National Conference of Social Media Processing (SMP) 2017, and ACM SIGSPATIAL 2020. He currently serves on the Editorial Board of ACM IEEE TRANSACTIONS ON SMART GRID (TSC), ACM Transactions on Intelligent Systems and Technology (TIST), ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT), GeoInformatica, and CCF Transactions on Pervasive Computing and Interaction (TPCI). In recent years, he was involved in the program or organizing committees of over 70 conferences and workshops.