IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 10 October 2022, accepted 24 January 2023, date of publication 1 February 2023, date of current version 8 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3241488

RESEARCH ARTICLE

Personalized Federated Learning for In-Hospital Mortality Prediction of Multi-Center ICU

TING DENG, HAZLINA HAMDAN, RAZALI YAAKOB, (Member, IEEE), AND KHAIRUL AZHAR KASMIRAN
Department of Computer Science, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang, Selangor Darul Ehsan 43400, Malaysia

Corresponding author: Hazlina Hamdan (hazlina@upm.edu.my)

This work was supported by the Fundamental Research Grant Scheme (FRGS), Ministry of Higher Education Malaysia, under Grant FRGS/1/2021/ICT02/UPM/02/2.

ABSTRACT Federated learning (FL), as a paradigm for addressing challenges of machine learning (ML) to be applied in private distributed data provides a novel and promising scheme to promote ML in multiple independently distributed healthcare institutions. However, the non-IID and unbalanced nature of the data distribution can decrease its performance, even resulting in the institutions losing motivation to participate in its training. This paper explored the problem with an in-hospital mortality prediction task under an actual multi-center ICU electronic health record database that preserves the original non-IID and unbalanced data distribution. It first analyzed the reason for the performance degradation of baseline FL under this data scenario, and then proposed a personalized FL (PFL) approach named POLA to tackle the problem. POLA is a personalized one-shot and two-step FL method capable of generating high-performance personalized models for each independent participant. The proposed method, POLA was compared with two other PFL methods in experiments, and the results indicate that it not only effectively improves the prediction performance of FL but also significantly reduces the communication rounds. Moreover, its generality and extensibility also make it potential to be extended to other similar cross-silo FL application scenarios.

INDEX TERMS Federated learning, non-IID, personalized, ICU, mortality prediction, electronic health records.

I. INTRODUCTION

With the promotion of electronic health record (EHR) systems, a huge amount of EHR data have emerged [1]. The EHR datasets, which contain exhaustive information such as patient diagnosis and treatment, underpin the application of machine learning (ML) in digital health. Moreover, its rich resources and valuable implicit information have also made ML one of the hottest technologies in its secondary analysis [2]. Nevertheless, due to the privacy and sensitivity of EHR, the application of traditional ML which refers to centralizing or releasing these data, poses not only legal, ethical, and regulatory challenges, but also technical ones [3]. Though there are some corresponding solutions to get around these restrictions, such as removing some key information to anonymize the patient data or adding privacy-preserving algorithms in the transmission process to prevent data leakage [4], the above problem has not been fundamentally solved because they still involve data migration.

Federated learning (FL) [5], [6], which emerged as a paradigm to address the concern of ML on private distributed data sources brings promising prospects to further promote ML in the digital healthcare field [7]. It is a distributed ML setting that can effectively assist multiple independent clients, such as mobile phones, IoT devices, and organizations, to conduct isolated data usage and ML modeling in accordance with user privacy protection, data security, and government regulations [8]. For healthcare, FL can implement ML in independent institutions without sharing any raw EHR data, which enables common and valuable information contained by the isolated data silos to be shared on the premise of protecting patient privacy and sensitive

The associate editor coordinating the review of this manuscript and approving it for publication was Antonio J. R. Neves.

information. In typical EHR applications, FL can help to find clinically similar patients across institutions to support medical research and applications [9], develop a general decentralized framework for prediction of hospitalization caused by cardiac events [10], as well as predict the mortality rate and stay time of ICU [11], including that under COVID-19 [12].

However, while FL has been proven to be effective and feasible in EHR from independent institutions, its performance can be degraded by the non-independently and identically distributed (non-IID) and unbalanced nature of these EHR data silos. Specifically, the non-IID feature can result in a significant reduction of model effectiveness in FL, like prediction accuracy loss of the clients' local ML models [13], [14], [15], and this situation can be magnified by the data unbalance [6]. Furthermore, the skewness of non-IID datasets (the divergence of IID data) also has a significant impact on FL performance. It can even be claimed that whether the validity of FL on non-IID data can be guaranteed depends on the extent to which the data distribution skew to non-IID [16]. Because when this skewness reaches a certain degree, the performance of the FL model will be affected, resulting in an accuracy-loss which increases with the growth of the skewness [17], [18]. Overall, for the application of FL in a multi-institution EHR scenario, the data distribution nature of non-IID and unbalance, especially that with high skewness, can reduce the model performance, even resulting in locally independent trained models exhibiting better performance than the FL-trained model, thus removing the main incentive of these healthcare organizations to participate in FL and even making FL meaningless.

To address the challenge outlined above, numerous FL optimization techniques have emerged, which have been summarized and divided into global optimization and local adaptation by D. Ting et al [19]. The local adaptation methods are specially proposed to deal with the statistical challenges in FL, which enables each participant to obtain a personalized model rather than accept a shared unified model. At present, personalized federated learning (PFL) incorporating an early straightforward "FL training + local adaptation" scheme and various subsequent techniques [20] has become a popular research branch [21]. As several personalized FL studies [20], [22] suggest, FL can recover from performance degradation by personalizing individuals' local models with their specific data when confronted with heterogeneous data environments such as non-IID and unbalanced distributions.

Consistent with the premise of PFL techniques, we argued that it is no longer applicable to generate a unified functional model for all FL participants in the non-IID and unbalanced data environment. Consequently, to cope with the challenge we propose a Personalized One-shot Local Adaptation (POLA) FL method after modifying the optimization problem of the standard FL. The proposed method aims to improve the performance of in-hospital mortality prediction in an actual multiple independent ICU center environment. Moreover, in order to further verify the effectiveness of the proposed method, we naturally divide the distributed ICU datasets in two different ways to generate ICU centers with different non-IID data skewness while preserving the actual data distribution. Experiments demonstrate that POLA can effectively enhance the model's mortality prediction performance in this data environment, as well as significantly reduce the number of communication rounds of FL training.

The main contributions included in this work are: 1) we underpinned our research problem by conducting experiments on baseline FL in the data context of this study. 2) we transformed the original global optimization problem of standard FL into a problem optimized for each individual, and then proposed a PFL method called POLA to generate highly personalized models for independent ICU centers. 3) we experimentally compared the POLA with baseline FL and two other PFL methods to demonstrate that it not only improves the model performance but also effectively reduces the communication overhead of FL.

The rest of this paper is organized as follows. Section II introduces the preliminary knowledge related to baseline FL, personalized FL, federated knowledge distillation, and AutoML. Section III presents the detailed designs of our proposed personalized FL scheme. The experimental evaluation and analysis are presented in Section IV. Finally, the work is discussed and concluded in Sections V and VI, respectively.

II. PRELIMINARIES

A. BASELINE FEDERATED LEARNING

The prototype and baseline of FL is a distributed ML algorithm based on mini-batch Stochastic Gradient Descent (SGD) named FederatedAveraging (FedAvg) [6]. Early optimization strategies for distributed ML generally involve iterative averaging of local models via adapting SGD in the local training process for optimization [23]. FedAvg is an adaptation of this kind of strategy under data privacy concerns. It is an orchestration pattern of distributed clients coordinated by a central server, where the clients both collect data and perform major computation tasks, and the central server coordinates the training process by integrating updated information exchanged with the clients [6].

The optimization objective of FedAvg can be defined as a global minimization problem below.

$$\min_{\omega\in\mathbb{R}} F(\omega), F(\omega) \overset{\text{def}}{=} \sum_{i=1}^{N} p_i f_i(\omega) \tag{1}$$

where N is the number of clients participating in the FL training, ω is a vector that contains global model parameters and $f_i(\omega)$ is the objective function of the i-th client which is determined by an arbitrary specific ML model and optimization algorithm. The optimization problem can thus be interpreted as figuring out optimal ω that can minimize the average loss over training models on all clients. p_i specifies the relative impact for the i-th client, which meets the conditions being $1 > p_i \geq 0$ and $\sum_{i=1}^{n} p_i = 1$. It is generally with two settings,

$p_i = 1/n$ or $p_i = d_i/d$, where d is the total data amount, d_i is that of client i.

To illustrate this method, its specific learning process and pseudo code are presented in Algorithm 1 [6]. The central server first establishes and initializes a global sharing model and then sends it to randomly selected clients. The selected clients independently and parallelly implement the SGD optimizer with pre-set local iterations and mini-batch data size on the receiving global model with their own unique data and then return the updated model or model parameters to the server. After receiving the information returned by all participating clients, the server updates the global model by performing a weighted average of these parameters according to the data proportions of each client. Again, the clients perform local training after receiving the updated global model and return their updated local model parameters to the server. These steps are repeated until a preset number of communication rounds is reached.

Algorithm 1 FedAvg Algorithm

```
Inputs:
- local training data on each client; - unified global model
Outputs:
- unified global model with updated parameters
Initialize:
- total communication rounds $R$; - local training iterations $E$;
- local mini-batch data size $B$; - learning rate $\eta$; - parameters
$\omega$ of global model
for each communication round $r$ form 1 to $R$
    Server update:
        Randomly select $N = C \times K$ clients, $C \in (0, 1]$, $K$ is total clients
        Send $\omega_r$ to all selected clients
        After all selected clients sending back updated $\omega_r^i$ do
        $\omega_r = \sum_{i=1}^{N} p_i \omega_r^i$
        Update $\omega_{r+1} \leftarrow \omega_r$
Client update:
  for client $i$ from 1 to $N$:
        Initialize the local model parameters $\omega_r^i \leftarrow \omega_r$
        Split local training data into batches of size $B$
        for each iteration $e$ from 1 to $E$:
            for each batch $d$:
                $\omega_r^i = \omega_r^i - \eta \nabla f_i(\omega_r^i, d)$ //$f_i$ is loss function
        end for
    end for
    return $\omega_r^i$ to server
  end for
end for
```

B. PERSONALIZED FEDERATED LEARNING

The initial intention of FL is to generate a globally unified model that performs effectively across the majority of participating clients. Since this idea has been proven to be limited in dealing with non-IID and unbalanced data [16], [17], personalized federated learning (PFL) has emerged as a compensation. Just as mentioned by Kulkarni et al. [20] and Tan et al. [21], the performance deterioration caused by heterogeneous data in FL can be addressed by personalized solutions.

Recently, research on PFL has set off a boom. There are numerous PFL strategies that have been developed to address the problem of the unified global model's failure to generalize well in FL while facing a data heterogeneity problem [21]. Since this study involves local adaptation to personalize FL, we also briefly summarize the related methods as follows: 1) Model fine-tuning. In highly heterogeneous data, performance gains can be achieved by simply fine-tuning all or part of the parameters of the global model obtained from FL training with private data locally on the client [18], [24]. 2) Local loss regularization. The client-drift problem caused by data heterogeneity is alleviated by adding regularization loss in the local training process to obtain better-performing personalized models [25], [26]. 3) Meta-learning. Its representative mechanism in FL is first to learn a parameterized model (or meta-learner) through the FL training process by algorithms like MAML and Reptile, then a specific personalized model for each client can be fast trained under the guidance of the meta-learner [27], [28]. 4) Multi-task learning aims to learn various models for multiple related tasks simultaneously, which is consistent with the mechanism of local adaptation for FL [29], [30]. 5) Transfer learning enables knowledge sharing among related domains to improve a learner's performance. In the FL setting under a heterogeneous data scenario, it helps the client models complete the local adaptation so as to get personalized models [31], [22]. 6) Knowledge distillation (KD) can be associated with FL to distill the knowledge like classification scores [32] and logit vectors [33] of the global model to guide the local client models in learning their personalized models.

Although all these PFL methods can improve the performance of FL on non-IID data problems, the ways in which they further personalize ML models are different. For example, model fine-tuning, meta-learning, multi-task learning, and transfer learning all personalize the parameters of the global model learned in FL. Local loss regularization personalizes the loss function of individual models in the FL learning process. Knowledge distillation can simultaneously personalize the structure and parameters of individual models as well as hyperparameters. This work aims to make the models as personalized as possible to gain performance enhancement as much as possible for FL. Therefore, the KD technique that has the most potential for model personalization is employed. In the next section, its related applications in FL are reviewed.

C. FEDERATED KNOWLEDGE DISTILLATION

KD is a student-teacher learning strategy with weak model correlation that was proposed and popularized by Hinton et al. [34]. It is extensively implemented in two major domains: model compression and knowledge transfer [35]. For model compression, KD can be used to learn a lightweight model with decent performance from the trained cumbersome model to meet the needs of real-time or edge applications. As to knowledge transfer, KD refers to a student-teacher learning structure in which the models that provide and learn knowledge are regarded as teacher and student, respectively. It enables students to learn from a larger pre-trained teacher model or an ensemble of teacher models. Consequently, KD is also regarded as an effective method that is frequently employed to transfer information from one network to another in ML.

Based on this knowledge transfer feature, KD has been applied to FL, and their combination is called federated knowledge distillation (FKD). In general FKD schemes, the global shared model is regarded as a teacher to guide the independent clients to train their local models [20]. Different from the standard FL method that directly exchanges models or parameters between clients and server, FKD allows distillated model knowledge to be exchanged as information. Thus, the communication cost during FL training can be significantly reduced, especially for deep ML models. However, since the distilled knowledge generally cannot contain as much information as the model parameters, FKD methods are usually accompanied by a decline in model accuracy.

A typical FKD method is federated distillation (FD) [33], which only exchanges the prediction logit vectors between server and clients to make the communication overhead model-independent. Compared with the baseline FL, it significantly reduced the training communication overhead but greatly decreased the model accuracy. Whereafter, a hybrid FD method (HFD) [36] is proposed as an enhancement of FD by adding an average covariate vector to the corresponding logit vectors. However, even though the model accuracy of HFD is improved compared with FD under the premise of constant communication cost, it is still lower than that of baseline FL. Although these approaches can reduce the communication cost in FL, the sacrifice of model accuracy is not worth the gain, especially in non-IID data, because participating individuals may not get any model performance gain in FL.

To alleviate this problem, some studies introduced public datasets in FKD. For instance, FedMD [32] pre-downloads a public dataset on each client to distill a classification score as exchanged knowledge. Another similar method is MHAT [37]. Each of its clients also holds a public dataset to generate the exchanged information. By introducing public datasets, both methods can reduce the communication cost while maintaining or improving the model accuracy. However, appending public datasets to FL is not recommended because it violates the original FL intention of not sharing raw data [21]. In addition, if all the clients need to download the public dataset frequently, there will be a sizable additional communication burden [15].

In addition to being utilized in FL to decrease communication overhead, KD can also be used to learn heterogeneous models for independent clients to deepen their personalization. This takes advantage of the KD's weak model correlation, which means that the teacher and student models aren't required to have the unified structure or set of hyperparameters. This extension in FL denotes that the local model of each independent individual can be regarded as a student model that is independent of the teacher model to learn high personalization according to the distribution characteristics of its local data. Li Hu et al. [37] conducted this strategy by generating heterogeneous models for clients while reducing communication overhead to compensate for the accuracy loss in FKD.

D. AutoML

This study utilized a heuristic algorithm involving automated machine learning (AutoML) in the optimization of personalized models, which may be confused with existing comparable studies. Thus, to show the difference between our proposed method and the existing "FL + AutoML" approaches, we conducted a retrospective analysis of related studies as follows.

AutoML is a combination of automation and machine learning (ML), booming in both academic and industrial fields in recent years. Its emergence has handed over the ML processes that require massive human interventions and efforts to the machine itself, such as algorithm and model selection, further realizing the real 'machine learning' [38].

Recently, more and more researchers have discovered that AutoML can be combined with FL to address the problem that the pre-defined unified model is not suitable for non-IID data distribution as FL has developed. Currently, the most popular use of AutoML in FL is neural architecture search (NAS), which is typically used for personalized design and optimization of clients' local models. For example, to save communication resources and accommodate edge devices in FL, Hangyu Zhu et al. [39] proposed an evolutionary real-time federated NAS approach that not only optimizes the performance of deep neural network (DNN), but also reduces the local payload of independent clients. Besides, a method named FedNAS [40] and a general framework named MGF-NAS [41] have also been developed for similar purposes to automate the model selection process in FL.

By reviewing the existing federated AutoML research, it can be found that almost all of them focus on the NAS of DNN models, especially convolutional neural networks (CNNs). Because the structure of the DNN model has a great impact on the communication overhead and the performance of FL, its automatic design and optimization can bring the most considerable benefits. But since our study does not involve DNN and is not limited to NAS, we do not compare it with existing federated NAS methods.

III. PROPOSED METHOD

A. PROBLEM DEFINITION

As can be observed from (1), standard FL is to optimize the parameters of the unified global model. However, after the experimental analysis in Subsection IV-D, it can be found that this optimization objective is no longer applicable in the data environment of this study. Therefore, we modified the optimization problem and expressed it as below:

$$F(\alpha,\theta) \overset{\text{def}}{=} \sum_{i=1}^{N} \min_{\alpha_i,\theta_i,\in\mathbb{R}} f_i(\alpha_i,\theta_i(\omega)) \tag{2}$$

where α and θ respectively represent the structure and parameters of local client model, ω is consistent with (1), which represents the parameters of the global model. This definition demonstrates how we changed the optimization problem of FL from determining the parameters for a unified global model into finding the optimal unique model structure and parameter sets for each independent individual in FL. Furthermore, it can also be seen that the specific parameters θ_i of each participant are related to the parameters ω of the global model, which means that the optimization problem of this work is not separated from the original FL setting, and its purpose is to further rebalance the global generalization experience with the local data knowledge to produce the optimal personalized models.

B. OVERALL FRAMEWORK

The proposed scheme is a two-step and one-shot PFL, the overview of which is illustrated in Fig. 1. *Two-step* here refers to FL training and local adaptation, where FL training is to obtain a shared model with adequate global generalization experiment, and local adaptation is a subsequent step to generate high-performance personalized models for independent individuals.

One-shot means the local adaptation only needs to be performed once for each individual in the entire training process. This one-shot adaptation process is a KD-based student-teacher learning, which regards the selected shared model as the teacher and treats locally independently personalized models as students. It enables the independent ICU centers to parallelly design their own personalized student models and then makes these student models learn from both the teacher model and their own datasets to improve performance by rebalancing global experience and local data knowledge.

Furthermore, in order to enable the student models to obtain the most suitable personalization design to optimize their performance, the adaptation step also includes an optimization process of the personalized model. However, this process is usually time-consuming and labor-intensive. To simplify and automate it, a classical heuristic technique - Genetic Algorithm (GA) is introduced. GA is a classical and effective evolutionary algorithm that searches for the optimal solution through selection, crossover, and mutation. In this study, it can simultaneously provide a wide search space and optimal solutions for hyperparameters and model structures that need to be designed automatically. The detailed content of the proposed method will be described in the next subsection.

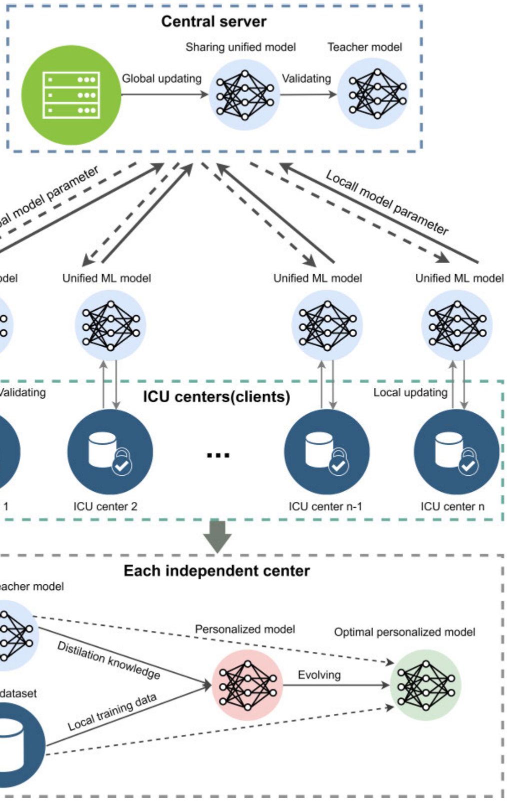


FIGURE 1. The illustration of the proposed scheme.

C. DETAILED DESCRIPTION

Algorithm 2 demonstrates the specific implementations of the proposed method POLA. As described above, step 1 is to complete the baseline FL training to obtain the teacher model required for subsequent local adaptation. The teacher model is critical to the outcome of the local adaptation. However, from the validation experiment in Section IV, we can see that the baseline FL is no longer unable to ensure the performance of its global model in the multi-center ICU data environment. If we directly take the global model obtained when training is completed as the teacher model, POLA's effectiveness cannot be guaranteed.

Therefore, in order to obtain a teacher model with stable performance and sufficient generalization knowledge, we adjust the baseline FL, the details of which are shown in Algorithm 3. We first divide the local training dataset of each center into validation and training data, and then use them for FL training and the global model's validation, respectively. Next, when the global shared model has learned enough generalization experience at a preset threshold number of training roundsR_w, the average validation error of all participants in each round is calculated to decide whether the current global model can be selected as the teacher model. Finally, when the entire FL training is over, the global model with the minimum validation error is selected.

Step 2 is performed in parallel on each independent ICU center, which mainly contains two procedures. Procedure 1 is to coordinate the entire local adaptation process by the

GA algorithm, which can automatically provide personalized solutions and evolve to produce the optimal one for each participant. Procedure 2 is to build and train the personalized model according to the solutions provided in the Procedure 1, and then return the results to evaluate.

In Procedure 1, the solution of the model personalization which involves structure design and hyperparameter selection and the inverse of model validation error which is returned by Procedure 2 are treated as chromosome and fitness function of GA, respectively. Then, GA will do selection, crossover, and mutation to evolve proper personalized solutions according to the fitness. When the evolutionary process is completed, the best solution will be the final personalized setting of the student model.

Algorithm 2 A Personalized One-Shot Local Adaptation FL Method - POLA

```
Inputs:
- same as Algorithm 1
Outputs:
- personalized model (p-model) with unique structure $\alpha_i$ and parameters $\theta_i$ for each center $i$
Initialize:
- parameters that Algorithm 3 needs; - scaling factor$\beta$; - local adaptation epochs $E_l$; - number of evolved generations $G$ and population size $P$
Step 1: FL training
  Complete the Algorithm 3 to get t-model
Step 2: Local adaptation
Each independent centre $i$ parallelly and locally does:
    receive t-model from server
    split local data into training data $D_t^i$ and validating data $D_v^i$
    Procedure 1: Produce personalized model solutions and evolve the optimal one
      generate the initial parent solutions $P_0$
      do Procedure 2 to fit $P_0$
      for$i$=1,2... $G$ do
        GA-operation $P_i$:
          do Procedure 2 to fit $P_i$
      end for
      Save the final personalized model corresponding to the highest-ranking solution
  Procedure 2: Train the personalized model
      build p-model with arbitrary structure $\alpha_i$ and hyperparameters via solutions from procedure 1
      initialize p-model 's parameters $\theta_i$
      for each iteration $e$ in $E_l$ do
        for batch $d_t \in D_t^i$ do
          $\theta_i = \theta_i - \eta_i \Delta L_i\left(\theta_i, \alpha_i, \beta, d_t\right)$
      end for
    end for
    $loss_{val} = L_i\left(\theta_i, \alpha_i, \beta, D_v^i\right)$
    return $loss_{val}$
```

In Procedure 2, each ICU center first independently builds its own personalized models according to the solutions provided by Procedure 1. Then the structured personalized models are initialized based on the teacher model that produced by Step 1. To make up for the drawback that student model usually can't outperform teacher model in KD scheme and speed up the training process, the initialization is layerwise. The input and the first hidden layer of the model are initialized directly as the corresponding parameters of the teacher model, and the remaining layers are initialized randomly. This is exactly what we need, because the base layer of the neural network model can contain more general knowledge.

Algorithm 3 Adjusted FedAvg Algorithm

```
Inputs:
- same as Algorithm 1
Outputs:
- teacher model t-model
Initialize:
- parameters that Algorithm 1 needs; - pre-set threshold training
round $R_w$
for $r = 1, 2, \ldots R$ do:
    Server update:
    $\boldsymbol{\omega}_{r+1}$=ServerUpdate($\boldsymbol{\omega}_r$)// in Algorithm 1
    receive $loss_{val}^i$ from centre $i$ in all $N$ centres
    if$r >= R_w$:
      $loss_{val} = \sum_{i=1}^{N} \frac{1}{N} loss_{val}^i$
      if $loss_{val}$ is minimum:
        t-model = g-$model_r$
Center update:
    for $i = 1, \ldots N$ do:
      $\boldsymbol{\omega}_r^i$=ClientUpdate($\boldsymbol{\omega}_r$,$D_t^i$)// in Algorithm 1
      $loss_{val}^i = l_k(\boldsymbol{\omega}_r^i, D_v^i)$
      return $\boldsymbol{\omega}_r^i$,$loss_{val}^i$ to server
    end for
end for
```

Next, the initialized models are going to learn from both the teacher model and the local dataset. It treats the general experience of the teacher model as soft target and the specific knowledge in the local raw data as hard target. To make the local personalized model learn as much knowledge as possible from the teacher model, we utilize two different methods to distill the outputs and features of the teacher model, respectively. The outputs distillation is a classical class probability distillation method [34], which tries to minimize the variance between the classification probability distributions of teacher and student. After estimating the classification probability of a neural network via a SoftMax function as (3) (where z_n represents the n-th category output in M objectives and T is the temperature factor which is used to control the weights of each soft target), if we express the last layer's prediction outputs of the teacher model and the student model as logit vectors z_t,andz_s respectively, then their divergence loss can be represented as l_{s1} in (4).

$$p(z_n, T) = \frac{exp(z_n/T)}{\sum_m exp(z_m/T)} \tag{3}$$

$$l_{s1} = L_R((p(z_t, T), p(z_s, T)) * T^2 \tag{4}$$

Generally, L_R is Kullback-Leibler (KL) divergence loss, but it can also be set to the Cross Entropy or MSE loss depending on the actual situation. What is specifically used in this study is the MSE loss function.

The feature distillation approach is to transfer knowledge from teacher to student by minimizing the divergence between the joint density probability estimations [42]. It first expresses the feature space of the teacher model and the student model as two conditional probability distributions $p_{n|m} \in [0, 1]$, $q_{n|m} \in [0, 1]$, and then uses KL loss to calculate the difference between them, and the training loss function l_{s2} shown below can be obtained.

$$l_{s2} = \sum_{n=1}^{M} \sum_{m=1, n \neq m}^{M} p_{n|m} \log(\frac{p_{n|m}}{q_{n|m}}) \tag{5}$$

As for the hard target, it is generally learned by the Cross Entropy loss function. Since this research is a binary classification task, a binary cross entropy loss function is adopted, which is symbolized by l_h as follows:

$$l_h = -\sum_{n=1}^{M} [y_n log\sigma(x_n) + (1 - y_n)log(1 - \sigma(x_n))], \sigma(x_n) = sigmoid(P(Y = 1|x)) \tag{6}$$

Finally, the training loss function L of the student model can be expressed as the follow:

$$L = \beta(l_{s1} + l_{s2})/2 + (1 - \beta)l_h \tag{7}$$

where $\beta\epsilon[0, 1]$ is a scaling factor to balance the local specific knowledge and global general knowledge. It can be seen that its value has a crucial effect on the performance of the personalized model. When it is large, the personalized student model learns more about the teacher model, and in turn, learns more about the local data.

IV. EXPERIMENTS AND ANALYSIS

A. DATA PREPROCESSING

The proposed scheme was developed in a multi-center ICU scenario which is based on an actual and freely available EHR database named eICU Collaborative Research Database, version 2.0 (eICU-CRD v2.0) [43]. This database is generated by teleICU, an actual project of Philips Healthcare, and collated by the Laboratory for Computational Physiology (LCP) at MIT. It comprises de-identified health data from over 200,000 admissions of more than 139 thousand unique ICU patients involving 335 units at 208 hospitals across the United States between 2014 and 2015 [44]. The eICU-CRD not only retains the natural characteristics of independently distributed data silos but also has abundant data resources that can properly support actual cross-silo FL application research.

Since the database is an unprocessed raw EHR, in order to obtain good research results, this study mainly refers to relevant benchmark research work [45] to do the variable selecting and preprocessing, which includes the following key steps.

1) SELECTING THE COHORT

This step is to filter the raw data based on criteria such as age range, number of records, and invalid key information, which results in 30,680 unique patients covering 1,164,966 records.

2) SELECTING THE VARIABLES

As shown in Table 1, this mortality prediction task selects 19 feature variables that reflect hospitalization status as inputs and 1 variable that indicates survival status as an output within a fixed time window of 48 hours for each patient.

3) VARIABLES PREPROCESSING

This process includes categorical variable encoding by one-hot encoding (OHE), numerical variable normalization, and input matrix padding. Finally, an input matrix of size 200*442 for each unique patient is obtained.

TABLE 1. Experimental variables of eICU-CRD in this work.

Input variables	Categorical	Eyes, Motor, Verbal, Admission diagnosis, GCS Total, Gender, Ethnicity
	Numerical	Height, Weight, Age, Systolic blood pressure, Heart rate, Mean arterial pressure, Diastolic blood pressure, O2, Glucose FiO2, Respiratory rate, Temperature, pH
Output variable	0 for alive and 1 for mortal	

B. DATA DISTRIBUTION

The research problem of this study involves not only the non-IID data but also its skewness. Thus, the data distribution involving how data is non-IID and how data skews to non-IID is crucial. Currently, the generation task of non-IID data is done artificially in most FL-related studies [6], [16], [37], which generally assign data evenly to each client based on different category labels and regulate the skewness to non-IID by the variance of data categories contained in the independent clients.

However, due to the lack of practical application support, this artificial way of generating data distribution not only fails to account for how real-world data distribution bias affects FL, but also ignores the unbalanced nature of real-world distributed datasets. Furthermore, the applicability of research results depends on how actually the experimental dataset simulates the distribution that will occur. Therefore, as mentioned by M. J. Sheller et al. [46], if feasible, an actual distribution that preserves the natural characteristics of the data is the best option for FL.

This study generates non-IID data in a natural way, which completely preserves the original distribution characteristics of eICU-CRD to simulate independent ICU centers with non-IID and unbalanced data. According to different non-IID skewness requirements, we naturally generate ICU centers according to hospital and ICU unit type, respectively. Together with the IID data distribution for comparison, this study finally includes the following three data distribution division ways:

1) IID AND EVEN DATA DISTRIBUTION

All datasets from the participating ICU centers are pooled, shuffled, and then evenly partitioned into the required number.

2) NON-IID AND UNBALANCED DATA DISTRIBUTION BASED ON HOSPITALS

The 208 hospitals in the ICU-CRD with varying numbers of patient cases are naturally treated as independent ICU centers. Since most of them have only a small number of patient admission records, we set a threshold at 600 to filter out those that cannot participate in FL training. Ultimately, 12 hospital-based ICU centers with a total of 9660 unique patient records are produced.

3) NON-IID AND UNBALANCED DATA DISTRIBUTION BASED ON ICU UNIT TYPES

In eICU-CRD, patients with different types of disease are admitted to corresponding ICU units, which results in greater variation in their related feature variables among different unit types.

TABLE 2. ICU unit types of eICU-CRD and their original patient amount.

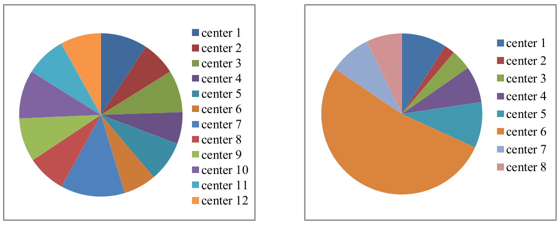
Unit type (n (%))	
Coronary Care Unit/Cardiothoracic ICU	15,290 (7.61)
Cardiac Surgery ICU	9,625 (4.79)
Cardiothoracic ICU	6,158 (3.07)
Cardiac ICU	12,467 (6.21)
Medical ICU	17,465 (8.70)
Medical-Surgical ICU	113,222 (56.37)
Neurological ICU	14,451 (7.19)
Surgical ICU	12,181 (6.06)

As shown in Table 2, all 335 ICU units with a total of 30,680 unique patient records are classified into 8 different types. Accordingly, we performed another data generation method according to the ICU unit types to increase the non-IID skewness of the data distribution. Finally, the database can be divided to simulate 8 independent unit-type-based ICU centers.

In addition to the natural generation methods of ICU centers to participate in FL, we also retained the original patient amount for each center. Fig. 2 shows the unbalanced amount distributions under two different non-IID data after cohort selecting.

C. EXPERIMENTAL SETTINGS

The proposed method is implemented in Python and all experiments are conducted on a computer with Intel 3.00GHz i7-9700 16GB CPU and NVIDIA GeForce RTX2060 6G GPU.



(a) Hospital-based ICU centers (b) Unit-type-based ICU centers

FIGURE 2. Patient amount distributions of two different non-IID data divisions.

1) MACHINE LEARNING MODEL

The specific ML model we employed in this work is Multilayer Perceptron (MLP), which has both unified and personalized designs. The unified design is applied in FedAvg with a fixed structure and pre-set hyperparameters. Specifically, it adopts a fixed model structure of two hidden layers with 100 nodes each and employs a rectified linear unit (ReLu) as the activation function. The optimizer is SGD with a momentum of 0.9 and the loss function is a binary cross entropy loss function.

As to the personalized design, the structure and hyperparameters of the MLP model are not fixed, and their specific values are determined by the evolutionary process. We restrict the structure of the model to two or three hidden layers and empirically provide the search space of the layer size and several influential hyperparameters, the detailed settings of which are shown in Table 3.

TABLE 3. Parameter space settings for the personalized model.

Parameter Name	Type	Searching space
Hidden layer	int	[2, 3]
Hidden layer size	int	[64, 80, 96, 112, 128, 144, 160, 176, 192, 218, 224, 240, 256]
Activation function	list	['relu', 'elu', 'tanh']
Learning rate	float	[0.0005 ~ 0.05]
Weight delay	float	[1.0e-03, 1.0e-04, 1.0e-05, 1.0e-06]
Batch size	int	[50, 70, 90, 110, 130, 150, 170, 190, 200]

As a result, the chromosome in evolutionary solutions is finally composed of four hyperparameters and three variables corresponding to the model structure, which are all real-encoded. In addition, we set the maximum training epoch of the personalized model to 20 and added an early stopping mechanism during the training process to prevent the model from overfitting. That is, the training will be terminated early when the validation error does not decrease continuously. Other than these, the other personalized model settings are the same as those of the unified design.

2) HYPERPARAMETER SETTINGS

In the first step, we set the proportion of clients participating in training C to 1.0, mini batch size B to 50, training iterations E to 5, learning rate η to 0.01, threshold training rounds R_w to 5, and the total number of communication rounds R to 100 (the training stop criterion). In the second step, we set the distillation temperature T to 10, the partition ratio of the training data D_t and the verification data D_v to 4:1.

As mentioned earlier, the scaling factor β has an important effect on the performance of the proposed method. We suggest its value should depend on the specific data distribution. When a well-performing teacher model is produced in slightly non-IID skewed data, the personalized models should learn more from the teacher model, but when the performance of the teacher model is degraded by highly non-IID data, the personalized models should be more biased towards the local datasets. Accordingly, we set its value in hospital-based and unit-type-based non-IID data to 0.6 and 0.4, respectively.

As to the evolutionary process, the population size and generations are set to be 20 and 5, respectively, which depends on the searching space and is also limited by the experimental conditions. The values of crossover and mutation operators are empirically set with probabilities of 0.9 and 0.1, respectively.

3) EVALUATION METRICS

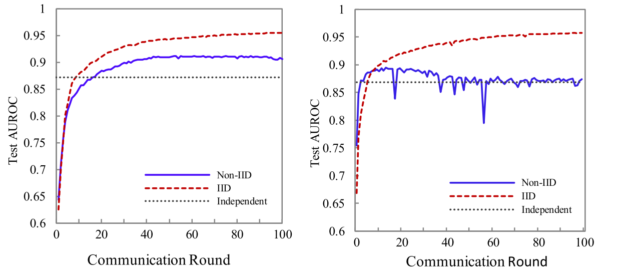
In accordance with [45], this work employed the Area Under the Receiver Operating Characteristic Curve (AUROC) to measure the mortality prediction results because the extreme unbalance of patient survival status has made the simple estimate of percentage accuracy meaningless. AUROC can well evaluate the performance of the prediction model in the case of unbalanced data classes and provide a basis for selecting the best prediction results. Furthermore, in order to truly reflect the performance of FL in non-IID and unbalanced data distribution, all our experiments are presented by the average of independent individual models' prediction results.

D. RESULTS ANALYSIS

1) THE IMPACT OF DATA DISTRIBUTION ON BASELINE FL ALGORITHM

In this subsection, we verified the impact of different data distributions that are described in Subsection B on the performance of baseline FL. Fig. 3 shows the mortality prediction results of FedAvg when the data distributions are IID and non-IID, as well as that of locally independent training. The locally independent trained model was treated as a benchmark to evaluate whether the FL-trained model achieved a performance gain for its participants.

The observation results show that different distributions of the same data have a great impact on the performance of FL. Compared with the even IID distribution, the naturally hospital-based and unit-type-based distributions both significantly degrade the performance of the baseline FL, and as the difference in data distribution increases, the performance degradation gets more obvious and even causes FL to fail to converge.



(a) Hospital-based ICU centers (b) Unit-type-based ICU centers

FIGURE 3. Compare results of FedAvg within 100 communication rounds.

Furthermore, it can be observed in Fig.3 (a) that the performance of FL-trained models in non-IID data distribution is obviously better than that of the local independent training models. This indicates that when the data distribution is not highly skewed to non-IID and unbalanced, the baseline FL can effectively improve the model performance. Nevertheless, in Fig.3 (b), we can see that, with the substantial increase in the non-IID and unbalanced characteristics of the data distribution, the baseline FL not only becomes unable to converge but can also hardly bring performance gains to the ML models.

These observations confirm the research problem of this work. That is, the prediction performance of FL can be degraded by the non-IID and unbalanced nature of data, and the higher the skewness of non-IID and unbalanced data, the more significant the performance degradation. In severe cases, locally independently trained client models can even outperform the FL-trained models, resulting in FL becoming meaningless. We argue that the cause of this problem might be that it is ineffective to obtain a unified working model for all participants from FL training in the heterogeneous data environment. In this data context, global collaboration without considering the unique characteristic of individuals usually cannot bring performance gains to most participants. Therefore, locally adapting and personalizing the FL-trained unified model on the independent client should be a good choice to tackle this problem.

2) COMPARISON EXPERIMENT

In this section, we compare the proposed method with the baseline FL and two other PFL methods to show that the proposed method works. The first comparable PFL method is a simple base + personalization layers local fine-tuning (FT) method [24], which is called FT-FedAvg in this paper. After receiving the FL global shared model, each independent individual freezes the base layer of the model and then updates the high-layer parameters with its local data for several epochs to gain personalization while maintaining the generalization knowledge in the high-layer parameters. Specifically, we utilize this method to fine-tune the global shared model of FedAvg for two epochs.

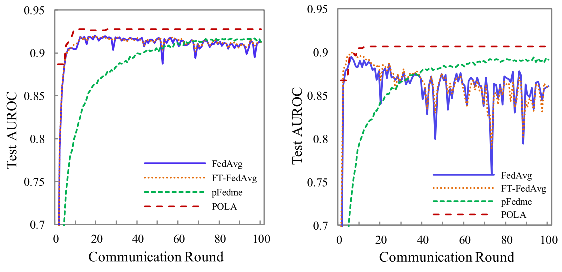
Another comparable PFL method is called pFedme [47] which personalizes FL by regularizing clients' loss functions with Moreau Envelopes. Its objective is also to balance personalization and generalization on each client to gain performance. To be fair, we selected the optimal parameter combination for pFedMe according to the data characteristics of this study. We first set several relevant parameters according to the requirements of the original pFedMe: personal learning rate $\eta = 0.001$, computation complexity $K = 5$, model additional parameter $\beta = 2$, $\lambda = 20$. Then we set the local training epochs to 50 and 80 for the hospital-based and unit-based data distributions, respectively, according to the characteristics of the amount of data in this study. This is because we found through experimental observation that the local training epochs of pFedMe should be appropriately increased with the increase of the client's local data amount, so as to ensure the convergence speed. In addition, other FL hyperparameters, such as the number of communication rounds, the number of participating clients, and the training data batch size, are consistent with the proposed method.

Fig. 4 shows the average prediction AUROC of the baseline FL and three PFL methods over 100 communication rounds. It should be noted that the initial setting of POLA is a one-shot local adaptation method. That is, the unique teacher model is found in the preset communication rounds and then adaptation is performed once to generate local personalized models. Here, in order to better demonstrate its performance, we adapt all the teacher models selected within the total of 100 rounds and the global model of the first round to generate personalized models for all participating ICU centers, thereby the corresponding curve of which is shown. For example, in a 100-round training, the global models selected for subsequent adaptation under the unit-type-based data distribution are in the 1st, 5th, 6th, 8th, 9th, 12th, and 13th rounds, respectively.

Furthermore, in order to present the performance of these methods in more detail, Table 4 shows the prediction results in the 5th and 100th communication rounds. The experiment was independently performed with different random seeds five times, and their average AUROCs with 95% confidence intervals are shown.

Besides, in order to observe the experimental results from the perspective of independent participants, the mortality prediction results of each ICU center's local model after 100 full rounds of training are shown in Fig. 5.

We can see from the overall findings shown in Fig. 4 and Table 4 that our proposed scheme POLA outperforms the other two PFL methods in both prediction performance and overall convergence rate under non-IID data distributions that have different skewness. From the individual perspective presented in Fig. 5, POLA also achieves acceptable performance, but its effectiveness varies depending on the distribution of the data. Compared with the best performing comparison method, pFedme, POLA significantly makes all unit-type-based ICU center models achieve performance gains, but only 58.33% of hospital-based ICU center models obtain performance enhancement. This indicates that POLA is more effective in the environment where the amount of independent dataset is sufficient and the non-IID skewness of the overall data is high.

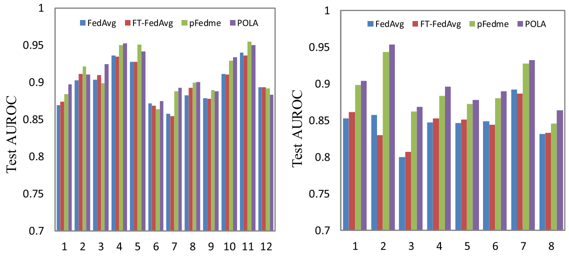


(a) Hospital-based ICU centers (b) Unit-type-based ICU centers

FIGURE 4. Overall mortality prediction results of four FL training methods in two different data distributions.

TABLE 4. The average predicted AUROC of four different FL schemes in the 5th and 100th communication round. the best results have been bolded.

	5th round		100th round	
	Hospital-based	Unit-type-based	Hospital-based	Unit-type-based
FedAvg	0.8876(± 0.0116)	0.8788(± 0.0101)	0.8977(± 0.0106)	0.8470(± 0.0111)
FT-FedAvg	0.8883(± 0.0117)	0.8871(± 0.009)	0.8990(± 0.0082)	0.8456(± 0.0161)
pFedme	0.7170(± 0.0264)	0.7114(± 0.0209)	0.9099(± 0.008)	0.8892(± 0.0065)
POLA	**0.8988(± 0.0136)**	**0.8897(± 0.0088)**	**0.9121(± 0.0128)**	**0.8982(± 0.0078)**



(a) Hospital-based ICU centers (b) Unit-type-based ICU centers

FIGURE 5. Individual mortality prediction results of four different FL training methods in two different data distributions.

FT-FedAvg, a simple adjustment of baseline FL, is greatly reliant on the global shared model, resulting in highly unstable performance. Overall, it achieves a certain performance gain over FedAvg. But from the standpoint of individual benefits, this improvement has no practical significance at all. As for pFedme, it can effectively overcome the obstacle of non-IID and unbalanced data to obtain stable and superior performance when the entire FL training is completed, but its convergence speed is too slow. As shown in Table 4, its performance gap with POLA at the 5th round of communication probably needs at least 30 subsequent training rounds to catch up, which requires a significant amount of computational and communication resources.

In conclusion, from the results of the comparative experiments, it can be seen that POLA not only effectively overcomes the non-IID and unbalanced data barriers with different skewness to generate personalized models with superior performance for each independent ICU center but can also significantly reduces the number of communication rounds in the FL training process, thus saving computational and communication overhead.

V. DISCUSSIONS

This section discussed several properties of the proposed method. The first one is compatibility. It can be seen from the experiment results that the effectiveness of the proposed method largely depends on the teacher model. This is also the reason why we need to select a well-performing teacher model during the FL training process. Intuitively, the better the obtained teacher model is, the better the generated personalized models are. But after experimental observation, we found that it is not the case. We speculate that this is due to POLA requires the teacher model not only to include generalization knowledge but also to be able to fit the parameter update direction of all student models. Therefore, although the proposed method seems to be a general FL training + local adaptation method, it is not compatible with arbitrary FL approaches.

Another property is extensibility, which involves two aspects: a) application scenario extension. Although POLA is especially proposed for predicting the mortality of inpatients in a multi-center ICU, it can also be applied to similar cross-silo scenarios. For example, the biomedical fields like disease incidence rate forecasting or medical image recognition, and the financial fields like multi-party borrowing detection. b) ML model extension. Although this study only employs the MLP model, it can also be extended to the application of other NN models, especially the DNN models. As model structure and hyperparameters have a greater impact on the performance of DNN, which can lead to higher performance gains. For example, if a lightweight DNN model is trained in FL and then tuned to more complex personalized models, considerable performance gain and communication overhead savings could be achieved.

VI. CONCLUSION

This study aims to enable FL to generate highly personalized ML models for each participant to tackle the predictive performance degradation in an actual multi-center ICU scenario. It keeps the natural and complete non-IID and unbalanced data distribution of the independent ICU centers, making it more significant for practical healthcare applications. We first studied the characteristics of the baseline FL in this data scenario to analyze the reason for its performance degradation. Then, we proposed POLA, a one-shot and two-step personalized scheme to make the performance of FL recover from non-IID and unbalanced data. POLA rebalances global experience and local data knowledge by making a one-shot adaptation for FL to produce a personalized local model for each independent ICU center. We experimentally demonstrate that it cannot only improve the performance of FL by generating superior-performing and highly personalized models but also significantly reduce the number of training communication rounds for FL.

REFERENCES

[1] B. Shickel, P. J. Tighe, A. Bihorac, and P. Rashidi, "Deep EHR: A survey of recent advances in deep learning techniques for electronic health record (EHR) analysis," *IEEE J. Biomed. Health Inform.*, vol. 22, no. 5, pp. 1589–1604, Sep. 2018, doi: 10.1109/JBHI.2017.2767063.

[2] J. Wu, J. Roy, and W. F. Stewart, "Prediction modeling using EHR data: Challenges, strategies, and a comparison of machine learning approaches," *Med. Care*, vol. 48, no. 6, pp. S106–S113, Jun. 2010, doi: 10.1097/MLR.0b013e3181de9e17.

[3] W. G. van Panhuis, P. Paul, C. Emerson, J. Grefenstette, R. Wilder, A. J. Herbst, D. Heymann, and D. S. Burke, "A systematic review of barriers to data sharing in public health," *BMC Public Health*, vol. 14, no. 1, pp. 1–9, Dec. 2014, doi: 10.1186/1471-2458-14-1144.

[4] A. Gkoulalas-Divanis, G. Loukides, and J. Sun, "Publishing data from electronic health records while preserving privacy: A survey of algorithms," *J. Biomed. Informat.*, vol. 50, pp. 4–19, Aug. 2014, doi: 10.1016/j.jbi.2014.06.002.

[5] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," 2016, *arXiv:1610.05492*.

[6] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Y. A. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2017, pp. 1273–1282.

[7] N. Rieke, J. Hancox, W. Li, F. Milletarì, H. R. Roth, S. Albarqouni, S. Bakas, M. N. Galtier, B. A. Landman, K. Maier-Hein, S. Ourselin, M. Sheller, R. M. Summers, A. Trask, D. Xu, M. Baust, and M. J. Cardoso, "The future of digital health with federated learning," *NPJ Digit. Med.*, vol. 3, no. 1, pp. 1–7, 2020, doi: 10.1038/s41746-020-00323-1.

[8] P. Kairouz. (Dec. 2019). *Advances and Open Problems in Federated Learning*. [Online]. Available: https://hal.inria.fr/hal-02406503

[9] J. Lee, J. Sun, F. Wang, S. Wang, C.-H. Jun, and X. Jiang, "Privacy-preserving patient similarity learning in a federated environment: Development and analysis," *JMIR Med. Informat.*, vol. 6, no. 2, p. e20, Apr. 2018, doi: 10.2196/medinform.7744.

[10] T. S. Brisimi, R. Chen, T. Mela, A. Olshevsky, I. C. Paschalidis, and W. Shi, "Federated learning of predictive models from federated electronic health records," *Int. J. Med. Informat.*, vol. 112, pp. 59–67, Apr. 2018, doi: 10.1016/j.ijmedinf.2018.01.007.

[11] L. Huang, A. L. Shea, H. Qian, A. Masurkar, H. Deng, and D. Liu, "Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records," *J. Biomed. Informat.*, vol. 99, Nov. 2019, Art. no. 103291, doi: 10.1016/j.jbi.2019.103291.

[12] A. Vaid et al., "Federated learning of electronic health records to improve mortality prediction in hospitalized patients with COVID-19: Machine learning approach," *JMIR Med. Informat.*, vol. 9, no. 1, Jan. 2021, Art. no. e24207, doi: 10.2196/24207.

[13] K. Chandiramani, D. Garg, and N. Maheswari, "Performance analysis of distributed and federated learning models on private data," *Proc. Comput. Sci.*, vol. 165, pp. 349–355, 2019, doi: 10.1016/j.procs.2020.01.039.

[14] A. Nilsson, S. Smith, G. Ulm, E. Gustavsson, and M. Jirstrand, "A performance evaluation of federated learning algorithms," in *Proc. 2nd Workshop Distrib. Infrastructures Deep Learn.*, Dec. 2018, pp. 1–8, doi: 10.1145/3286490.3286559.

[15] H. Zhu, J. Xu, S. Liu, and Y. Jin, "Federated learning on non-IID data: A survey," *Neurocomputing*, vol. 465, pp. 371–390, Nov. 2021, doi: 10.1016/j.neucom.2021.07.098.

[16] K. Hsieh, A. Phanishayee, O. Mutlu, and P. Gibbons, "The non-IID data quagmire of decentralized machine learning," in *Proc. Int. Conf. Mach. Learn.*, vol. 119, Nov. 2020, pp. 4387–4398. [Online]. Available: http://proceedings.mlr.press/v119/hsieh20a/hsieh20a.pdf

[17] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-IID data," 2018, *arXiv:1806.00582*.

[18] T. Yu, E. Bagdasaryan, and V. Shmatikov, "Salvaging federated learning by local adaptation," 2020, *arXiv:2002.04758*.

[19] D. Ting, H. Hamdan, K. A. Kasmiran, and R. Yaakob, "Federated learning optimization techniques for non-IID data: A review," *Int. J. Adv. Res. Eng. Technol.*, vol. 11, no. 12, pp. 1315–1329, 2020. [Online]. Available: https://iaeme.com/MasterAdmin/Journal_uploads/IJARET/VOLUME_11_ISSUE_12/IJARET_11_12_125.pdf, doi: 10.34218/IJARET.11.12.2020.125.

[20] V. Kulkarni, M. Kulkarni, and A. Pant, "Survey of personalization techniques for federated learning," in *Proc. 4th World Conf. Smart Trends Syst., Secur. Sustainability (WorldS)*, Jul. 2020, pp. 794–797, doi: 10.1109/WorldS450073.2020.9210355.

[21] A. Z. Tan, H. Yu, L. Cui, and Q. Yang, "Towards personalized federated learning," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Mar. 18, 2022, doi: 10.1109/TNNLS.2022.3160699.

[22] Q. Wu, K. He, and X. Chen, "Personalized federated learning for intelligent IoT applications: A cloud-edge based framework," *IEEE Open J. Comput. Soc.*, vol. 1, pp. 35–44, Feb. 2020, doi: 10.1109/OJCS.2020.2993259.

[23] S. Zhang, A. Choromanska, and Y. Lecun, "Deep learning with elastic averaging SGD," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2015, pp. 685–693.

[24] S. K. Pye and H. Yu. (Aug. 2021). *Personalised Federated Learning: A Combinational Approach*. [Online]. Available: https://ui.adsabs.harvard.edu/abs/2021arXiv210809618K/abstract

[25] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic controlled averaging for on-device federated learning," in *Proc. 37th Int. Conf. Mach. Learn.*, 2019, pp. 5132–5143.

[26] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst. (MLSys)*, 2018, pp. 1–22. [Online]. Available: https://www.researchgate.net/publication/329734586

[27] Y. Jiang, J. Konečný, K. Rush, and S. Kannan, "Improving federated learning personalization via model agnostic meta learning," 2019, *arXiv:1909.12488*.

[28] M. Khodak, M.-F. Balcan, and A. Talwalkar, "Adaptive gradient-based meta-learning methods," 2019, *arXiv:1906.02717*.

[29] V. Smith, C. K. Chiang, M. Sanjabi, and A. Talwalkar, "Federated multi-task learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4424–4434.

[30] L. Corinzia, A. Beuret, and J. M. Buhmann, "Variational federated multi-task learning," 2019, *arXiv:1906.06268*.

[31] D. Gao, Y. Liu, A. Huang, C. Ju, H. Yu, and Q. Yang, "Privacy-preserving heterogeneous federated transfer learning," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2019, pp. 2552–2559, doi: 10.1109/BigData47090.2019.9005992.

[32] D. Li and J. Wang, "FedMD: Heterogenous federated learning via model distillation," in *Proc. NeurIPS*, Oct. 2019, pp. 1–8.

[33] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Communication-efficient on-device machine learning: Federated distillation and augmentation under Non-IID private data," in *Proc. NIPS*, Nov. 2018, pp. 1–6.

[34] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[35] L. Wang and K.-J. Yoon, "Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 6, pp. 3048–3068, Jun. 2021, doi: 10.1109/TPAMI.2021.3055564.

[36] J.-H. Ahn, O. Simeone, and J. Kang, "Wireless federated distillation for distributed edge learning with heterogeneous data," in *Proc. IEEE 30th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2019, pp. 1–6, doi: 10.1109/PIMRC.2019.8904164.

[37] L. Hu, H. Yan, L. Li, Z. Pan, X. Liu, and Z. Zhang, "MHAT: An efficient model-heterogenous aggregation training scheme for federated learning," *Inf. Sci.*, vol. 560, pp. 493–503, Jun. 2021, doi: 10.1016/j.ins.2021.01.046.

[38] Q. Yao, M. Wang, Y. Chen, W. Dai, Y.-F. Li, W.-W. Tu, Q. Yang, and Y. Yu, "Taking human out of learning applications: A survey on automated machine learning," 2018, *arXiv:1810.13306*.

[39] H. Zhu and Y. Jin, "Real-time federated evolutionary neural architecture search," *IEEE Trans. Evol. Comput.*, vol. 26, no. 2, pp. 364–378, Apr. 2022, doi: 10.1109/TEVC.2021.3099448.

[40] C. He, E. Mushtaq, and J. Ding, "FedNAS: Federated deep learning via neural architecture search," in *Proc. ICLR*, 2022, pp. 1–11.

[41] Z. Pan, L. Hu, W. Tang, J. Li, Y. He, and Z. Liu, "Privacy-preserving multi-granular federated neural architecture search a general framework," *IEEE Trans. Knowl. Data Eng.*, vol. 4347, pp. 1–12, 2021, doi: 10.1109/TKDE.2021.3116248.

[42] N. Passalis and A. Tefas, "Learning deep representations with probabilistic knowledge transfer," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 268–284.

[43] T. Pollard, A. Johnson, J. Raffa, L. A. Celi, O. Badawi, and R. Mark, "eICU collaborative research database v2.0," PhysioNet. [Online]. Available: https://physionet.org/content/eicu-crd/2.0/, doi: 10.13026/C2WM1R.

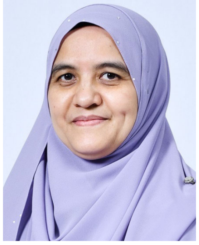
[44] T. J. Pollard, A. E. W. Johnson, J. D. Raffa, L. A. Celi, R. G. Mark, and O. Badawi, "The eICU collaborative research database, a freely available multi-center database for critical care research," *Sci. Data*, vol. 5, no. 1, pp. 1–13, Sep. 2018, doi: 10.1038/sdata.2018.178.

[45] S. Sheikhalishahi, V. Balaraman, and V. Osmani, "Benchmarking machine learning models on multi-centre eICU critical care dataset," *PLoS ONE*, vol. 15, no. 7, pp. 1–14, 2020, doi: 10.1371/journal.pone.0235424.

[46] M. J. Sheller, B. Edwards, G. A. Reina, J. Martin, S. Pati, A. Kotrotsou, M. Milchenko, W. Xu, D. Marcus, R. R. Colen, and Spyridon Bakas, "Federated learning in medicine: Facilitating multi-institutional collaborations without sharing patient data," *Sci. Rep.*, vol. 10, no. 1, pp. 1–12, 2020, doi: 10.1038/s41598-020-69250-1.

[47] C. T. Dinh, N. H. Tran, and T. D. Nguyen, "Personalized federated learning with Moreau envelopes," in *Proc. Adv. Neural Inf. Process. Syst.*, Jun. 2020, pp. 1–12.

TING DENG received the B.Sc. and M.Sc. degrees from the Wuhan University of Technology, Wuhan, China, in 2012 and 2014, respectively. She is currently pursuing the Ph.D. degree with the Department of Computer Science, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, with a focus on non-IID data issue of federated learning.



HAZLINA HAMDAN received the Ph.D. degree in artificial intelligence from the University of Nottingham, U.K., in 2013. She is currently a Senior Lecturer with the Department of Computer Science, Universiti Putra Malaysia. She has led several research projects funded by the Ministry of Higher Education (MOHE) and research university grant scheme (GP), and is currently leading two on going research projects, while being a co-researcher in three other projects. Her research interests include intelligent computing and applications, such as medical prognostics, pattern recognition, prediction systems, and optimization.

RAZALI YAAKOB (Member, IEEE) received the bachelor's and M.Sc. degrees in computer science from Universiti Putra Malaysia (UPM), in 1996 and 1999, respectively, and the Ph.D. degree from the University of Nottingham, U.K., in 2008. Currently, he is an Associate Professor/Lecturer at the Faculty of Computer Science and Information Technology, UPM. He is also the Dean of the Faculty. His research interests include artificial neural networks, pattern recognition, and evolutionary computation in game playing. He is a member of the Intelligent Computing Group at the Faculty.

KHAIRUL AZHAR KASMIRAN received the Ph.D. degree from The University of Sydney, Australia, in 2012. He is currently a Senior Lecturer with the Department of Computer Science, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Malaysia. His research interests include deep learning, reinforcement learning, performance engineering, formal verification, and software development.