Received: 2 June 2020 | Revised: 21 July 2020 | Accepted: 12 August 2020

DOI: 10.1002/spe.2894

SPECIAL ISSUE PAPER

WILEY

Towards efficient federated learning-based scheme in medical cyber-physical systems for distributed data

Kehua Guo[1] | **Nan Li**[1] | **Jian Kang**[2] | **Jian Zhang**[1]

[1]School of Computer Science and Engineering, Central South University, Changsha, China

[2]Department of Dermatology, Third Xiangya Hospital, Central South University, Changsha, China

Correspondence
Jian Zhang, School of Computer Science and Engineering, Central South University, Changsha, China.
Email: csu_jianzhang@263.net

Funding information
111 project, Grant/Award Number: B18059; Earmarked Fund for China Agriculture Research System; Fundamental Research Funds for the Central Universities of Central South University, Grant/Award Number: 2019zzts962; National Science Foundation of Hunan Province, China, Grant/Award Numbers: 2019JJ20025, 2019JJ40406; Natural Science Foundation of China, Grant/Award Numbers: 61502540, 61672535, 61977062

Summary

In recent years, Cyber-Physical Systems (CPS) and Artificial Intelligence (AI) have made good progress in the medical field. The medical CPS (MCPS) based on AI can realize the efficient and reasonable utilization of medical resources and improve the quality of medical process. However, current MCPS are still facing several challenges, and the privacy protection of medical data is one of the most critical challenges. Since medical data is stored in different hospitals, most studies collect data from decentralized hospitals to train a disease diagnosis model, which is not conducive to the privacy protection of patients. And in some existing solutions, it is also difficult for doctors to select the optimal model from multiple models in clinical diagnosis. In this paper, we propose a novel scheme based on federated learning in MCPS for training disease diagnosis models from distributed medical image data. Our scheme is divided into three parts: the model provider, the server, and the consumer, and a detailed working process is designed for each part. This scheme can not only effectively solve the problem of privacy protection, but also solve the problem of model selection for doctors and save storage space. It can ensure that consumers automatically get a steadily improved disease diagnosis model. This scheme is performed on simulated distributed medical image datasets. The experimental results show the effectiveness and superiority of our scheme.

KEYWORDS

computer-aided diagnosis, distributed data, federated learning, medical cyber-physical systems, model fusion

1 | INTRODUCTION

Cyber-Physical Systems (CPS) and Artificial Intelligence (AI) are two influential technologies in computer science, which have been widely applied in various fields, including intelligent industrial system,[1,2] intelligent transportation system,[3,4] intelligent medical system,[5,6] etc. As a breakthrough in emerging technologies, CPS and AI provide the ability to obtain and process large amounts of data from the physical world. The application of these two technologies to the medical industry is a potential research topic. Medical CPS (MCPS) refer to intelligent network systems that can continuously collect medical data from various medical devices, enabling doctors to make accurate medical diagnoses for patients based on the data.[7] AI can quickly become an expert in the medical field by analyzing and learning medical data, so as to assist

doctors in diagnosis. MCPS based on AI is a new design concept in clinical control. The use of AI to analyze the data collected in MCPS can assist the clinical diagnosis and reduce the work stress of doctors, thus providing higher quality services for the medical process, realizing the efficient and reasonable utilization of medical resources, and promoting the development of medical industry.

In the current MCPS, most studies use deep learning technology to make decisions and have achieved excellent results. For example, Long et al[8] developed a prototype diagnosis and treatment system CC-Cruiser for congenital cataract screening and put the classification model on the cloud server to assist doctors to complete the diagnosis. However, it requires a large amount of data to get a disease classification model by using deep learning technology. Generally, the amount of data collected from MCPS in a single hospital cannot meet the requirements, so it is necessary to collect data from multiple decentralized hospitals and upload them to the central server. After the central server collects the data, it uses deep learning technology to train a disease diagnosis model to improve the decision making of MCPS and assist doctors in diagnosis. When the data is collected to the central server, the hospital loses the ownership of the patient data, and cannot control the storage, use and flow of the patient data, which increases the risk of the disclosure and abuse of the patient information. These data are usually private and sensitive. Once these data are leaked, it will have a serious impact on the lives of patients, and even bring great losses to the lives, health and property of patients. Therefore, it is the current research hotspot that how to train a disease diagnosis model which can improve the decision making of MCSP by using the disease data collected by MCPS from distributed hospitals under the condition of privacy protection. Guo et al[9] have realized the privacy protection of patients and designed an intelligent medical diagnosis framework. The main principle is to place the training process of the model on the local hospital so that the data can be stored on the local hospital, and only the trained model can be uploaded to the cloud server for doctors to download. In practical applications, doctors need to select the appropriate model after preliminary diagnosis. However, due to some other limitations such as the lack of professionalism of the doctor, the doctor cannot choose an optimal model among many disease classification models.

Federated learning[10] provides ideas for solving these problems. Federated learning is a distributed machine learning method, which separates model training from cloud storage. It is able to derive a model from the rich data which stored on distributed devices without storing it centrally. Therefore, it is very appropriate to solve the problem of model training in the case of distributed medical data by using federated learning, which can well protect the patient privacy and solve the doctors' difficulties in model selection. However, in the existing researches, the application of federated learning in CPS is still relatively few, especially in MCPS. There are no researches that use federated learning method to improve the decision making of MCPS.

In this paper, we propose a novel scheme based on federated learning for training disease diagnosis models from distributed medical image data to solve the problems in MCPS mentioned above. The scheme takes into account the model provider, the server, and the consumer. In our scheme, the model providers, that is, the decentralized hospitals, collect medical data by the medical device in MCPS and train the corresponding diagnostic models locally by using the deep learning method. And then the model providers only upload these models to the server respectively, while the medical data is still stored locally in these hospitals, which can avoid the privacy disclosure caused by collecting the data centrally in MCPS. In the server, the model fusion method is used to fuse these local models into an aggregated model, which can be downloaded and used by the consumer to provide accurate diagnostic assistance and solve the problem of model selection for the consumer. In addition, we have also designed a verification process for the model update on the consumer, which can automatically provide steadily improved diagnosis results for the consumer, so as to reduce the work stress of doctors and provide higher quality service for medical process. Our main contributions are summarized as follows.

(1) This paper proposes a novel scheme based on federated learning for training disease diagnosis models from distributed medical image data to improve the decision-making of MCPS, which is divided into three parts: the model provider, the server and the consumer, and a detailed working process is designed for each part.
(2) This scheme can train a disease diagnosis model without centralized data storage, which not only reduces the risk of patient privacy disclosure, but also solves the difficulty for doctors to choose an optimal model and saves storage space.
(3) The incremental update method designed for the consumer in this scheme can ensure that the consumer automatically obtain a steadily improved disease diagnosis model, which make MCPS more consumer-friendly.

The remainder of this paper is organized as follows. Section 2 briefly reviews the related work. Section 3 describes the workflow of this paper and introduces the related technology and its theoretical basis. Section 4 indicates the experimental results and analysis. Section 5 presents the conclusion and describes future work.

2 | RELATED WORK

CPS is a multidimensional complex system which integrates communication, computation, and physical processes. It is divided into three parts: perception layer, network layer, and application layer. The perception layer is mainly responsible for collecting data, the network layer mainly provides real-time network services for the system, and the application layer mainly analyzes the data from the perception layer, and presents the corresponding results to customers. In the traditional clinical medical scene, doctors play the role of application layer, while medical devices play the role of perception layer. If we use AI to analyze the data collected in the perception layer to assist doctors in decision-making, medical resources can be used more efficiently and higher quality can be provided for medical process.

In recent years, with the development of deep learning technology, AI has bid farewell to the traditional if-then-else mode and has been widely studied and applied in computer-aid diagnosis.[11] In the field of computer-aided diagnosis, Liu et al[12] used the multiple kernel learning-based classifier and patient personal data to design an automatic glaucoma diagnosis architecture. Havaei et al[13] proposed a fully automatic brain tumor segmentation method based on deep neural network and realized a double-channel structure that is suitable for the identification of glioblastomas in magnetic resonance imaging (MRI). Esteva et al[14] used the classical convolutional neural network (CNN) model in deep learning to classify skin lesions. The trained CNN model performed as well as all medical experts tested in distinguishing keratinocyte carcinomas versus benign seborrheic keratoses, and malignant melanomas versus benign nevi. However, whether or not these studies have been used or are being studied, due to the limitations of the dataset size and disease characteristics, most of them focus on a single disease, and the research result obtained can only be used on that disease. In addition, most of these studies focus on improving the effect of auxiliary diagnosis, while ignoring the privacy protection of data. They usually collect the data to a central server and then process the data and training a model centrally, which is very unfavorable for protecting patient privacy. Once the data is transmitted to the central server, the hospital cannot control the use and flow of data, and the risk of patient information being leaked and abused of will increase.

Many researchers have tried to overcome these shortcomings and proposed improvements. In the view of the problem that most studies can only be used to build a single disease diagnosis model, Gibson et al[15] separated data loading, data augmentation, network architectures, and encapsulated them into independent modules, so as to facilitate researchers to quickly build neural network models for different types of medical images. However, this study still used a centralized data processing method, which is not conducive to protecting the personal data privacy of patients. In the field of medical big data, the commonly used technologies to protect the privacy of patients mainly include access control-based processing technology,[16] data encryption-based processing technology,[17] and anonymity-based processing technology,[18] etc. However, these technologies have disadvantages such as poor flexibility, large computational overhead and vulnerability to link attacks. Guo et al[9] proposed a medical-assisted diagnosis platform in the form of cloud services,[19] which only transmitted the trained disease model to the server, and without data transmission, so as to avoid the privacy leakage caused by data transmission. The doctor could choose the appropriate model to download and use. Although this study can effectively protect the privacy of patient, when there are too many models on the platform and the patient's disease is more complicated, doctors cannot choose a suitable model for computer-aided diagnosis, and there are also disadvantages such as high cost of storage space.

The federated learning allows the distributed devices to cooperatively learn a shared model, which can effectively solve the model selection problem for doctors. All training data is stored on the devices to ensure data security. Due to the short time proposed for federated learning, there are currently few kinds of research related to federated learning. These existing researches on federated learning are mainly divided into two aspects: technology and application. In terms of technology, Konečný et al[20] proposed structured updates and sketched updates to reduce the cost of an uplink communication in order to improve the communication efficiency between the client and the server. Nishio et al[21] extended federated learning and proposed a new FedCS protocol, which solved the client selection problem with resource constraints and accelerated the performance improvement in the model. These studies focus on optimizing the performance of federated learning, and do not solve the actual problems encountered in AI-assisted MCPS decision-making, that is, privacy protection and model selection. In terms of application, Chen et al[22] applied federated learning to the recommendation system and presented a federated meta-learning framework that locally trains the user-specific recommendation model through a shared parameterization algorithm. Brisimi et al[23] referred to the idea of federated learning and proposed an iterative cluster Primal-Dual Splitting algorithm for solving the large-scale sparse support vector machine (sSVM) problem dispersedly. This method has been applied to predict the future hospitalization of patients with heart-related diseases using electronic medical record data from various data sources. For these application researches, only a few researches are applied in the

medical field, and most of them focus on the analysis of medical text data, and the training of disease diagnosis model is relatively simple. In the use process, there is no complete process of design and too much improvement.

In this paper, the idea of federated learning is applied to our scheme. Under the condition that the data of each hospital is not collected centrally, an excellent model can be learned from these decentralized data, which can assist doctors in diagnosis.

3 | METHODOLOGY

In this section, we first describe the work process of our scheme. Then, we detail the three key steps in our scheme.

3.1 | Work process

In this paper, we use the idea of federated learning to solve the problem of model training when the data is distributed in various top hospitals. Figure 1 shows the architecture and work process of our scheme. The architecture of this scheme is mainly composed of three parts: (1) The model provider. It is mainly a client machine placed in a top hospital with rich medical data and responsible for providing data and training models. (2) The server. It is the central controller, and it is responsible for the collection, fusion and storage of models. (3) The consumer. The consumers are some hospitals that lack medical resources and need a computer-aided diagnosis.

In this scheme, there are three key steps: (1) A model provider first collects medical image data and uses these data to retrain the global aggregated model which is downloaded from the server to obtain a new local computer-aided diagnosis model, which is the medical image classifier. All model providers must use the same structure of the deep neural network

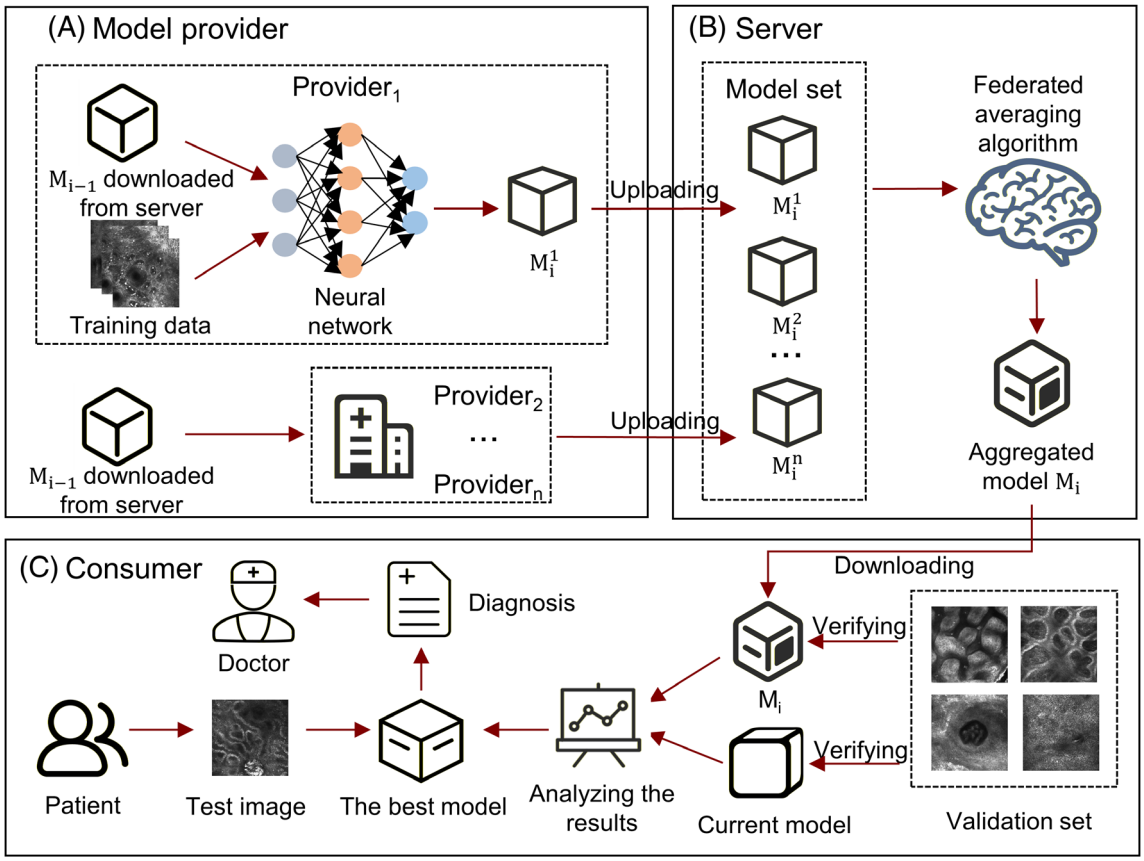

FIGURE 1 The architecture and work process of our scheme. (A) Model provider; (B) Server; (C) Consumer [Color figure can be viewed at wileyonlinelibrary.com]

for training in order to ensure the smooth process of model fusion on the server. At the same time, the initialization weights of the deep neural networks also must be the same. There is no limit to the amount of medical image data. (2) The model providers upload the new local models to the server, respectively. The data used to participate in model training is not uploaded to the server, which reduces the risk of privacy leakage. Then, the server uses the federated averaging algorithm to merge these new local models into an aggregated model. (3) The consumer downloads the aggregated model from the server, and verify the aggregated model on the validation set. If the model is validated, it can be put into practical use. The validation set is managed by the consumer and can be updated based on actual usage. The verification process is to ensure that the model currently used by the consumer is the optimal model.

The above process can be iterated continuously. During these iterations, the model provider uses the new medical data and the new global model aggregated in the last iteration to train a new local model, that is, each model provider downloads the latest aggregated model from the server and updates the latest aggregated model to its own initial model, and then trains a new local model with its own local new medical data. Similarly, there is no limit to the amount of new medical data. And these new local models are merged into a new global model. This global model not only retains the old knowledge learned before, but also learns new knowledge, which makes the performance of the global model continuously improve.

3.2 | Model training on distributed model providers

In the scenario of this paper, multiple distributed model providers need to collect data and train the models by themselves. Due to the limited data that a single model provider can provide, an appropriate neural network structure is very important for the model provider.

The current mainstream neural network structures, such as VGGNet,[24] Inception,[25] ResNet,[26] have shown good performance in the field of image classification. When choosing a suitable neural network model for the model provider, we need to consider the accuracy and the efficiency of training. Generally, a deeper network model means that better results can be achieved. But in fact, simply superimposing convolutional layers to increase the depth of the network cannot improve the accuracy of the model, or even make the model worse. The deeper networks are often accompanied by the vanishing gradient problem or the exploding gradient problem.[27,28] The classification performance of VGGNet will decrease if the number of layers is added after 19 layers, and VGGNet is not efficient due to the large number of channels in the convolution layer. Inception uses a dense structure to approximate a sparse CNN, and its accuracy and training speed are better than VGGNet. ResNet uses the residual idea to maintain a strong increase in accuracy when the depth increases. Its accuracy is higher than that of VGGNet and Inception, and its computational efficiency is also very efficient. It can save hardware resources for each model provider. Therefore, we choose ResNet to a train local model for the model provider.

ResNet is mainly composed of residual blocks, and Figure 2(A),(B) shows the residual block structures for two different components. In the residual block, the output of the previous layer is directly passed to the later layer by building the identity mapping, which can accelerate the training of ultra-deep neural network very quickly, and the accuracy of the model is also greatly improved. There are five main forms of ResNet: ResNet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet-152. The difference between the five networks mainly due to the difference in the residual block parameters and the number of intermediate convolutions. Considering that deeper networks will incur higher memory costs when training and storing models, we exclude ResNet-101 and ResNet-152 with more layers and chose ResNet-50 with a relatively high classification accuracy among the remaining ResNet-18, ResNet-34, and ResNet-50 to train a computer-aided diagnosis model for the model provider.

At the model provider, we collect medical image data, and preprocess the data, and then train a computer-aided diagnosis model with ResNet-50. During training, we remove the last layer of ResNet-50 and add a full connection layer with 128 neurons and an output layer. The modified network structure is shown in Figure 2(C). The categorical-crossentropy loss is used as the loss function.[29] The formula of the loss function is

$$L = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m} t_{i,j} \log p_{i,j}(w), \tag{1}$$

where n is the number of samples, m is the number of categories to be predicted by the network, $t_{i,j}$ represents the target output of the ith sample in the jth category, and $t_{i,j} \in \{0, 1\}$. $p_{i,j}$ represents the predicted output of the ith sample in the jth

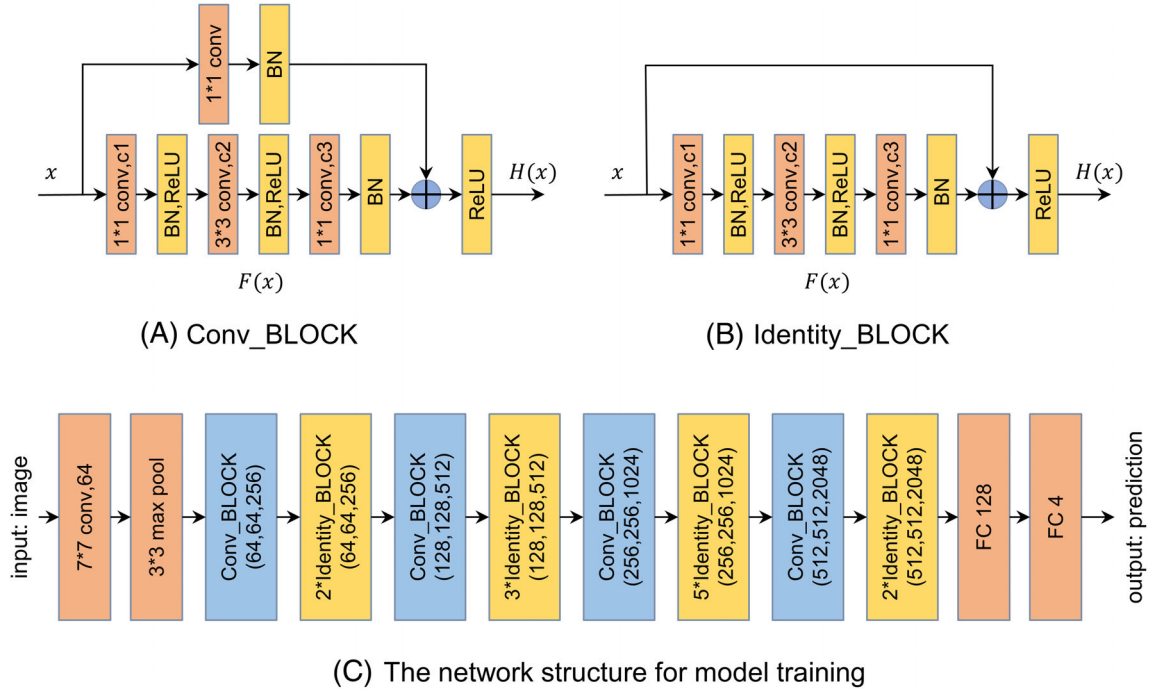

FIGURE 2 (A) Conv_BLOCK and (B) Identity_BLOCK are two different residual blocks. (C) is the network structure used in model providers [Color figure can be viewed at wileyonlinelibrary.com]

category made with model parameters w, and $p_{i,j} \in [0, 1]$. The model training algorithm at the model provider is shown in Algorithm 1.

Algorithm 1. Model training algorithm

Input: Model provider: k, current training round: t, epochs: E, local dataset: P_k, learning rate: α.
Output: Local model: w_t^k

download the aggregated model w_{t-1} from server
$w_t^k \leftarrow w_{t-1}$
$n_t^k \leftarrow \mid P_k \mid$ //n_t^k is the number of samples in the dataset of the client k
for epoch $i \in [1, E]$ **do** //train model by using SGD
    for batch $b \in P_k$ **do** //b is the sample set selected from P_k randomly
        $w_t^k \leftarrow w_t^k - \alpha \nabla L$
    end for
end for
send w_t^k, n_t^k to server
return w_t^k, n_t^k

During the training process of the model provider shown in Algorithm 1, the model provider first downloads the last round of the aggregated model from the server and uses it as the initial model for the local model. Then we trained the local model with the local dataset in model provider by using stochastic gradient descent (SGD) algorithm. Due to the relatively strong fitting ability of ResNet, a pre-trained ResNet can converge when training local models on the datasets provided by different model providers. In addition, the model provider with small number of images may get a model with poor classification accuracy. Since the final focus of our scheme is on the effect of the aggregation model, when we aggregate the model, we will give a smaller contribution to the model with a small number of images, so we do not need to pay too much attention to the classification accuracy of the local model. After the training is completed, the new local model and the sample number of the local data set participating in this training process are transmitted to the cloud

server. The time complexity of Algorithm 1 is $O\left(E \times \frac{|P_k|}{|b|} \times T_1\right)$, where T_1 is the time required for a batch to perform a gradient update.

3.3 | Federated averaging algorithm

After the server collects the computer-aided diagnosis model with parameters w_t^k transmitted from the model provider, the server needs to merge these models $w_t\_set = \{w_t^1, \ldots, w_t^K\}$ into a better model with parameters w_{t+1}. Therefore, an effective model fusion method is very important for our scheme.

The federated averaging algorithm proposed by McMahan et al[10] can effectively merge the models in w_t_set in each round to get a new aggregated model with parameters w_{t+1}. Its working mechanism is similar to the synchronous mode in the parallel training mode of the deep neural network.[30,31] All devices simultaneously get the parameters of the initial model, and the back-propagation algorithm is run on different devices to obtain the model gradients based on the training data of each device. Then, the gradients on different devices are weighted averaged according to the number of samples. And finally, the model is updated according to the averaged gradient. In fact, this method of weighted average the gradients is equivalent to weighted average the weights of the model.[10] This method is computationally effective. And this method can train the model without collecting data, which can effectively protect the privacy of model providers. In addition, Izmailov et al[32] also presented that for points on the SGD trajectory, that is, the weights saved on the SGD training trajectory, averaging can achieve better generalization than conventional training.

In this paper, we use the federated averaging algorithm to aggregate these models. The flow chart of the federated averaging algorithm is shown in Figure 3. First, a model provider k reads the same initial model w_t from server and then calculates the gradient g_k by using the local dataset P_k. And the formula for updating the local model on the model provider k is

$$w_{t+1}^k = w_t - \alpha g_k, \tag{2}$$

where α is learning rate, and $g_k = \nabla L_k(w_t)$.

Then, the server aggregates these local models from models providers and computes the final aggregated model as follows

$$w_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k, \tag{3}$$

where K is the number of model provides, n_k is the number of images provided by the kth model provider, n is the sum of the images provided by all model provider.

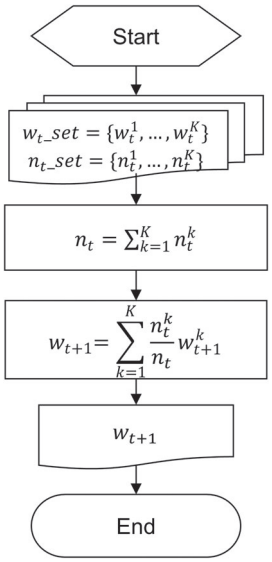

FIGURE 3 The flow chart of the federated averaging algorithm

The key step of this algorithm is to performing a weighted average of the sample number on the collected models $w_t\_set = \{w_t^1, \dots, w_t^K\}$, and after averaging, the aggregated model w_{t+1} is obtained. The federated averaging algorithm is shown in Algorithm 2.

Algorithm 2. Federated averaging algorithm

Input: The number of model providers: K, current training round: t
Output: Aggregated model: w_{t+1}
if $t = 0$
  w_t is a pretrained model on ImageNet dataset
for client $k \in [1, K]$ **in parallel do**
  $(w_t^k, n_t^k) \leftarrow$ Algorithm 1(k, t)
  add w_t^k, n_t^k into set M, N
end for
$n_t \leftarrow \sum_{k=1}^{K} n_t^k$ //n_t is the sum of the samples in round t
initialize w_{t+1}
for $k \in [1, K]$ **do**
  $w_{t+1} \leftarrow w_{t+1} + \frac{n_t^k}{n_t} w_t^k$
end for
return w_{t+1}

In Algorithm 2, an aggregated model is obtained by weighted averaging these parameters of these local models from model providers, and then the aggregated model can be downloaded and used by the consumer. In the special case $t = 0$, the aggregated model on the server is a pretrained model on the ImageNet dataset. The time complexity of Algorithm 2 is $O\left(E \times \frac{|P_k|_{\max}}{|b|} \times T_1 + T_2\right)$, where $|P_k|_{\max}$ is the maximum number of local dataset provided by each model provide, and T_2 is the time of model fusion.

3.4 | Incremental update model

In this section, we describe the incremental training process for the aggregated model and how the consumer verifies and updates the disease diagnosis model.

The training process of Algorithm 1 and Algorithm 2 can be combined to obtain an aggregated model, and this process is defined as federated training. Federated training can have multiple training rounds, and it can be iterated continuously to improve the performance of the aggregated model. During the iteration process, the server distributes the last round of the aggregated model w_{t-1} to each model provider, and each model provider uses the aggregated model w_{t-1} as its initial model. Then, the model provider uses the newly collected medical data P_k and w_{t-1} for training to obtain a new local model. The server collects these new local models and fuses them to obtain a new aggregated model w_t by using the Algorithm 2. After the above process is finished, the new round of training is completed.

In our scheme, each round of the aggregated model is updated by using new local models which are trained with new local data and last round of aggregated model. This method does not need to add the previous data into the new data to rejoin the training, which can save a lot of time. However, due to the influence of noise data and other factors in the training dataset, the accuracy of the aggregated model may not always increase. Therefore, in order to ensure the accuracy of the computed-aided diagnosis model used by consumers, we reserve some labeled data as a validation set and design a verification process for the updating process on the consumer side. When updating the model on the consumer side, we first compare the accuracy of the current model on the consumer side with the new aggregated model downloaded from the server on the validation set, and the model with the higher accuracy is updated to the current model. If the accuracy of the model exceeds the accuracy threshold used, the model can be used for computer-aided diagnosis. The federated optimization and verification process are shown in Algorithm 3.

Algorithm 3. Incremental update algorithm

Input: The number of model providers: K, validation set stored on consumer: V, current model stored on consumer: w_{cur}, the accuracy of w_{cur} on V: acc_cur, accuracy threshold for using: th
Output: computer-aided diagnosis: w_{cur}

for round $t = 1, 2, \ldots$ **do**
  $w_{t+1} \leftarrow$ Algorithm 2(K, t) //run on the server
  //run on the consumer
  consumer download w_{t+1} from the server
  calculate the accuracy acc_new of w_{t+1} on V
  if $acc\_new > acc\_cur$
    $w_{cur} \leftarrow w_{t+1}$
    $acc\_cur \leftarrow acc\_new$
  end if
  if $acc\_cur > th$
    use w_{cur} for computer-aided diagnosis
  end if
end for
return w_{cur}

In Algorithm 3, the server aggregates these local models trained with the new data to obtain a new aggregated model. And then the consumer downloads the new aggregated model and verifies it. Only the validated aggregated model can be used for assisting the doctor in diagnosis. The time complexity of the Algorithm 3 is $O\left(R \times \left(E \times \frac{|P_k|_{\max}}{|b|} \times T_1 + T_2 + V.size\right)\right)$, where R is the number of rounds currently trained, and V. size is the size of the validation set, representing the time spent by model w_{t+1} in validating on the validation set.

4 | EXPERIMENTS

Our scheme is universal and effective no matter how many models or large datasets they are. In order to verify the effectiveness of our scheme, we collect the skin Reflectance Confocal Microscopy (RCM) image data from the Third Xiangya Hospital of Center South University for experiments and take this data set as an example for experiments. Other types of disease are also applicable. These RCM data are real and effective. The experimental dataset contains images of four types of diseases: hypopigmentation, hyperpigmentation, demodicosis, and papillomatous hyperplasia. There are 800 samples for each disease, a total of 3200 samples. The resolution of each sample is 1000*1000, and the sample features are clear. The space occupied by each sample is about 200 KB. We randomly select 100 samples from each category of disease as the validation set, 100 samples as the test set, and the remaining 600 samples as a training set. Therefore, there are 2400 samples in the training set, 400 samples in the verification set and 400 samples in the test set.

These experiments are implemented in an Intel Xeon CPU @2.00GHz and NVIDIA Titan_Xp 12G GPU, using Python 3.6.4 and Keras 2.2.2. It is necessary to clarify that although we implemented our experiments on a machine equipped with a Titan GPU, our scheme is not limited by the GPU platform, and our scheme can also be implemented on a machine without GPU. Since GPU can accelerate the training speed of deep neural network model, the machine without GPU may need more time to train a model than the machine with GPU under the same training parameters. In practical application, this time is within our tolerable range.

In the experiment, we simulate the actual medical application scenarios and restore the data distribution and actual data flow of different hospitals as realistically as possible. We use the Accuracy (ACC) and the Area Under Receiver Operating Characteristic (ROC) Curve (AUC) to evaluate the effectiveness and superiority of our proposed Scheme. ACC is defined as the ratio of the number of samples correctly classified by the classifier to the total number of samples for a given test dataset. Generally, the higher the accuracy, the better the classifier. The value of AUC is the area under the ROC curve. AUC represents the probability that a positive example is ranked before a negative example. Generally, the higher the AUC value, the better the classification effect of the model.

4.1 | Confusion matrix analysis for the local models and the aggregated model

In this experiment, we simulate the actual fusion situation and output the confusion matrix of each model for analysis. First, we set the number of model providers as 3 and divide the 2400 samples into three parts in a random way, so as to restore the data distribution and actual data flow of different hospitals as realistically as possible. Because most hospitals are general hospitals now, people have no obvious preference for the choice of hospitals, which is random, especially for some common diseases. From a macro perspective, the disease data in each hospital is also randomly distributed from a large center. Therefore, the random distribution method we adopted can effectively restore the data distribution in most hospitals, so as to ensure the generalization ability of the aggregate model. The number of samples in each part is also randomly assigned, respectively: 675, 759, 966. Then, these three training data sets are assigned to the model providers. These model providers, respectively, use their own training data set, the ResNet-50 and the pretrained ImageNet model for training to obtain the local models. The batch size of training is set to 50 and the epoch of training is set to 30. Finally, the federated learning algorithm is used to obtain the aggregated model. The confusion matrix of each local model and the aggregated model on the test set are shown in the Figure 4.

In Figure 4, the rows of the confusion matrix represent the actual disease categories and the columns represent the disease categories predicted by the model. We number the four diseases hypopigmentation, hyperpigmentation, demodicosis, and papillomatous hyperplasia as 0, 1, 2, and 3, respectively. The value of the cell a_{ij} represents the number of samples of true category i predicted to be j, and the diagonal represents the number of samples predicted to be correct. The ruler on the right of Figure 4 shows the corresponding relationship between the value of cell and the color. The larger the value in the cell, the darker the color. The confusion matrix can display the classification results of the model intuitively, which is convenient to compare, analyze and evaluate the advantages and disadvantages of the model.

These confusion matrixes in Figure 4 show the classification results of the models intuitively. It can be seen that local models A, B, and C have both shown a poor classification result on a certain category of disease. For example, the number of samples predicted correctly on the 0 diseases by model A is only 53. The number of samples predicted correctly on the

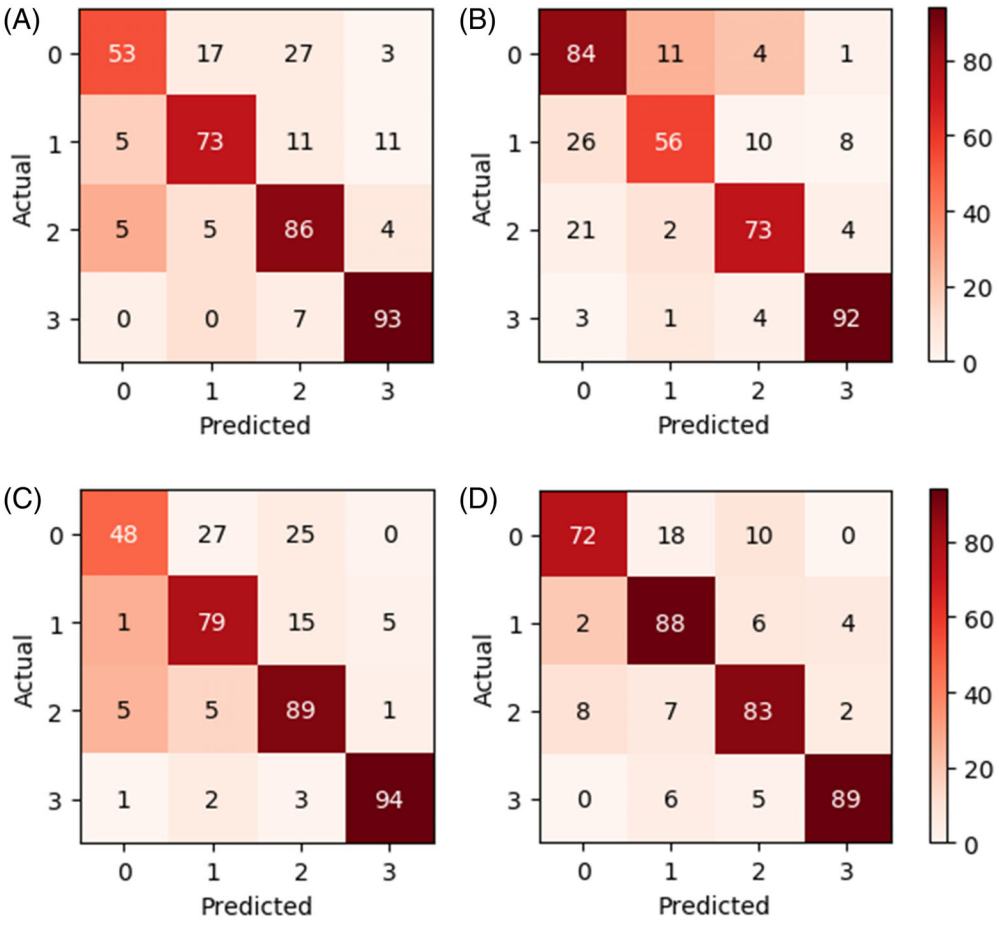

FIGURE 4 Test set confusion matrixes for computer-aided diagnosis models. (A), (B), and (C) are confusion matrixes for local models, and (D) is the confusion matrix for the aggregated model [Color figure can be viewed at wileyonlinelibrary.com]

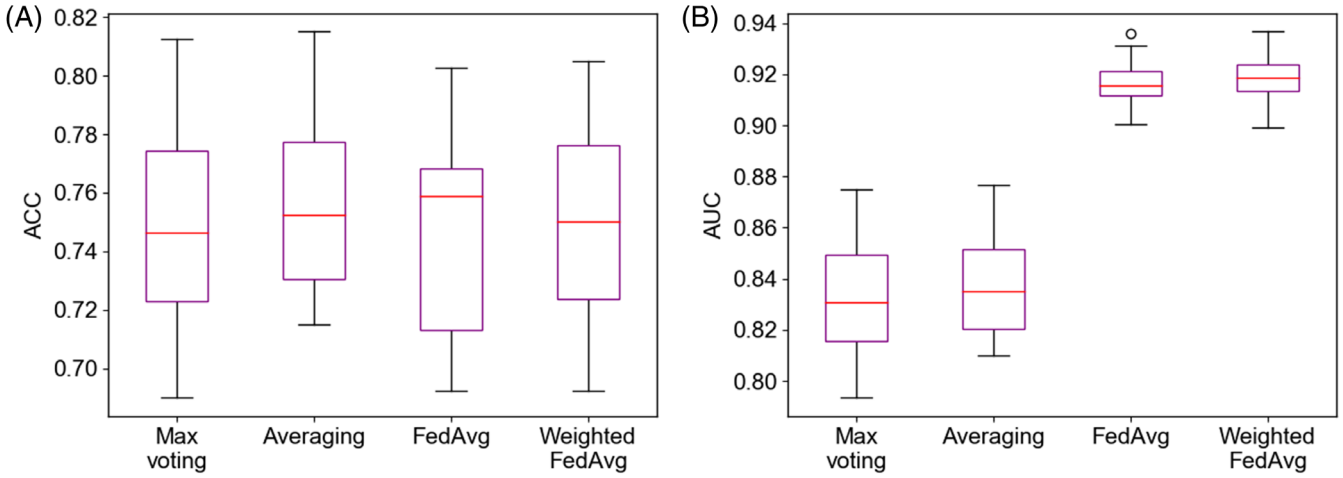

FIGURE 5 The distribution of the (A) Accuracy and (B) Area Under Receiver Operating Characteristic Curve values of the aggregated models fused by voting, averaging, FedAvg, weighted FedAvg [Color figure can be viewed at wileyonlinelibrary.com]

1 disease by model B is only 56, and the number of correct predictions for model C on the 0 disease is only 48. However, the aggregated model D performs better in the classification of various diseases without any major disadvantages, and the number of correct predictions in each category can reach more than 72. In addition, we observed that the accuracies of the aggregated model D on some diseases are not high as that of other local models on these diseases, such as the 3 disease. But this problem is not very important because we focus on the overall effect rather than the effect of single recognition. Each local model has very serious shortcomings in a single recognition, while the aggregated model is very balanced in the recognition of each category, and the overall recognition effect has been greatly improved. This shows that the model D aggregated by models A, B, and C can aggregate the advantages of models A, B, and C, and abandon the disadvantages of models A, B, and C. The overall effect of the aggregated model is improved compared to the local models, which is 5.5% higher than the local model with the highest accuracy. This illustrates the effectiveness of the fusion method based on the weighted average of training set size.

4.2 | Comparison of our method and other fusion methods

In this experiment, we compare the four fusion methods of voting, averaging, federated averaging algorithm without weighting samples (FedAvg) and federated averaging algorithm (weighted FedAvg) by using ACC, AUC, inference time and model size evaluation indicators. The inference time refers to the time it takes to infer a sample, and the model size refers to the storage space consumed by the model in each method. Voting is a method based on the prediction results of each local model, which choose the most voted category as the final prediction result. Averaging gets the prediction result by averaging the prediction results of each local models. FedAvg is a fusion method that directly averages the weights of local models. Weighted FedAvg perform a weighted average algorithm on the weights of local models according to the training set size of the local model.

In order to analyze the influence of the four fusion methods on the ACC and AUC values of the aggregated model, we train 10 groups of local models with different numbers, and then fused the local models in the 10 groups with the four methods, respectively. Figure 5 shows the distribution of the ACC and AUC of the aggregated models fused by these four fusion methods in the form of a box diagram. In order to ensure the fairness of this experiment, the same local models are used in each fusion method.

We can see from Figure 5 that there is no obviously difference on the ACC value between the models aggregated by the four fusion methods, and the all fluctuate in the range of 0.68 to 0.81. But on AUC indicator, the models aggregated by FedAvg and weighted FedAvg methods are significantly better than the models aggregated by voting and averaging methods. The AUC values of the models aggregated by FedAvg and weighted FedAvg are both above 0.9, while the AUC values of the models aggregated by voting and averaging methods range from 0.79 to 0.87. The AUC values of the models aggregated by FedAvg and weighted FedAvg methods are always about 0.03 to 0.14 higher than that of voting and

Method	Inference time	Model size
Max voting	$nT_a + T_b$	nM
Averaging	$nT_a + T_c$	nM
FedAvg	T_a	M
Weighted FedAvg	T_a	M

TABLE 1 The inference time and model size of the aggregated model by using different method

averaging methods. Generally, the higher the value of AUC, the higher the probability value of positive samples determined by the classifier than that of negative samples. This indicates that the models aggregated by FedAvg and weighted FedAvg methods can give the prediction result more definitely and illustrates the superiority of FedAvg and weighted FedAvg. In addition, the results in Figure 5 also show that FedAvg and weighted FedAvg perform comparable on ACC, while weighted FedAvg performs slightly better on AUC than FedAvg, and there is an outlier in FedAvg. Therefore, the weighted FedAvg is recommended for training disease diagnosis models, because weighted FedAvg takes into account the influence of training set size on the model and the performance of weighted FedAvg is more stable in the model fusion process.

Next, we analyze and compare the inference time and model size of these four fusion methods. In order to eliminate the influence caused by the performance differences of different machines, we use a proportional method to display the results. Table 1 shows the inference time and model size of the aggregated model by using four fusion methods.

As shown in Table 1, T_a represents the time required for a local model to test a sample, while T_b and T_c, respectively represent the time required for voting and averaging to process the predicted results of each local model. M represents the amount of storage required to store a local model. n represents the number of local models participating in this fusion. We can see from Table 1 that FedAvg and weighted FedAvg perform better than voting and averaging both in inference time and model size. Considering that the model aggregated by voting and averaging methods need to using the local models to predict the test samples and then giving a final diagnosis result for the test samples, the aggregated models obtained in these two methods are actually multiple models, so the storage space is n times the model size of a local model, while FedAvg and weighted FedAvg directly average or weighted average the weights of multiple local models, and the result of the fusion is only one model. Therefore, the storage space of the model aggregated by voting or averaging is n times that of the model aggregated by FedAvg or weighted FedAvg. Similarly, the inference time for voting or averaging includes the inference time for each local model and the time to process the results. While the inference time for FedAvg or weighted FedAvg is only equivalent to the inference time of a local model, which saves a lot of time and space.

In conclusion, among the four fusion methods, FedAvg and weighted FedAvg perform better than voting and averaging in AUC, inference time and model storage, which indicates the superiority of FedAvg and Weighted FedAvg.

4.3 | Effectiveness of our scheme

In this experiment, we simulate the actual data stream from the hospital, and continuously use the new data to train models for fusion to evaluate the performance of our scheme in practical applications. Similarly, the random distribution method can effectively restore the data distribution in most hospitals, so as to ensure the generalization ability of the aggregate model. We set up three model providers in this experiment. In each round of federated training, we randomly choose 120 samples to allocate to the three model providers, each getting 40 samples. In this case, the number of training sets for each model provider in each round is the same, that is, each model provider in each round uses the newly distributed 40 samples combined with the download aggregated model to train new local models. During the training process of local model, the batch size is 10, and the epoch is 50. Local models are transmitted to the server after training, and the models on the server is fused by federated averaging algorithm to get the aggregated model. The aggregated model of this round will serve as the initial model for the next federated training round. The consumer validates the new aggregated model before using it. We output the validation set ACC and AUC of the local models and aggregated models in each round to evaluate the performance of our scheme in practical applications. In addition, we also output the ACC and AUC on the consumer test set of aggregated models and that of the best model selected by

validation process in each round to illustrate the necessity of the validation process. The specific results are shown in Figures 6 and 7.

As shown in Figures 6 and 7, the federated learning algorithm is very effective. In most federated training rounds, the ACC and AUC values of the aggregated model are higher than the ACC and AUC values of the local models. This shows that after federated averaging algorithm, the aggregated model can learn from the strengths of local models and make up for the weaknesses of local models, so that the performance of the aggregated model is no less than that of the local models. The detailed analysis is described in Section 4.1. For some rounds which the values of ACC and AUC are slightly reduced after fusion, such as round 6, the possible reason is that there may be noise or new data distribution in the training samples of this round, resulting in the reduced effect of the aggregated model.

We can see from Figures 6 and 7 that the validation set ACC and AUC values of the aggregated model change from a lower value to a higher value and then to a fluctuation state with the increase of training rounds. The fluctuation range of the ACC value is 0.73 to 0.81, and the fluctuation range of the AUC value is 0.89 to 0.95. In the practical application of the consumer, if the aggregated model is directly used, the performance of the computer-aided diagnosis model at the consumer will be unstable. As shown in the Figures 6 and 7, the curve change of the ACC and AUC values of the aggregated model on the test set with the federated training round is similar to the fluctuation state of the ACC and AUC values of the aggregated model on the validation set. They are also from low to high and then go into a fluctuation state with the increase of training rounds. However, for the model selected after the model verification process, the ACC and AUC values on the test set show a relatively stable growth state. This shows that the validation process is necessary, and it can ensure the disease diagnosis model with good stability used in the consumer, so that the consumer can have a better use experience.

In addition, in each federated training round of this experiment, all the local models are trained with new data, and the data used in the previous training rounds do not participate in new training round. But the ACC and AUC values of both the aggregation model and the model on the consumer showed an increasing state with the increase of the training rounds, that is, the increase of data, and gradually stabilized. This shows that our scheme can balance the relationship between new knowledge and old knowledge. While learning new knowledge, it can keep most of the knowledge learned before, and the old knowledge learned before does not need to be relearned, so it has the ability to process the increasing data. With the increase of training data, the performance of the model at the consumer can be improved continuously.

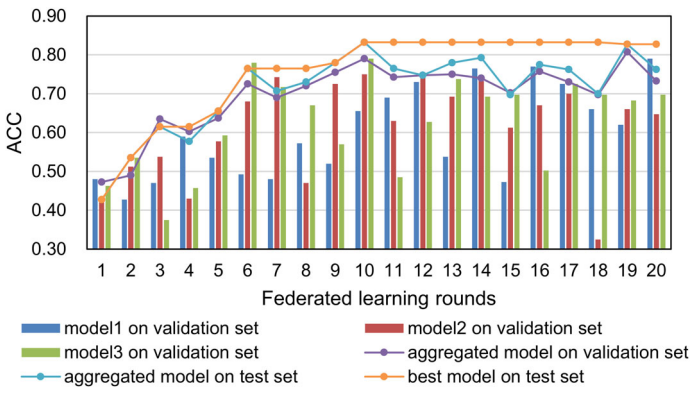

FIGURE 6 Accuracy versus federated training rounds for the local models, the aggregated model and the best model [Color figure can be viewed at wileyonlinelibrary.com]

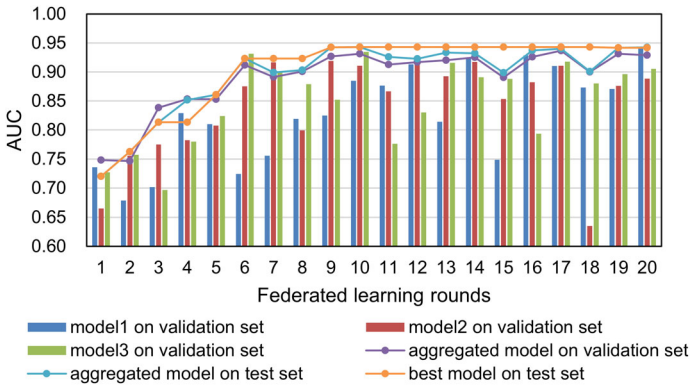

FIGURE 7 Area Under Receiver Operating Characteristic Curve versus federated training rounds for the local models, the aggregated model and the best model [Color figure can be viewed at wileyonlinelibrary.com]

4.4 | Influence of hyperparameters on aggregated model

Before the training of the deep neural network model, it is necessary to select a set of optimal hyperparameters for the training of this model, so as to improve the performance and effect of this model. But these hyperparameters are often set according to experience, so there may be subjectivity. Therefore, in this experiment, we study the influence of the two important hyperparameters, the learning rate η and the epoch E on the performance of the aggregated model during the federated training.

In the experiment of exploring the impact of the η on the aggregated model, we set the learning rate η from 10^{-2} to 10^{-6} in turn, while other parameters, the training process and the fusion process remain unchanged. Figure 8 shows the influence of different learning rates on ACC and AUC values of the aggregated model.

We can see from Figure 8 that it is appropriate to set the learning rate as 10^{-4} or 10^{-5} for the training of the model provider. These two cases can make full use of the data, and the aggregated model can achieve good performance in fewer federated training rounds. Moreover, in the same federated training rounds, that is, the round 20, the aggregated model with $\eta = 10^{-4}$ and $\eta = 10^{-5}$ show better performance than the aggregated model with other learning rates on the ACC and AUC and the ACC and AUC value could be as high as 0.4 and 0.3, respectively. Among them, the aggregated model with $\eta = 10^{-4}$ can achieve better performance faster than the aggregated model with $\eta = 10^{-5}$, but as the number of federated training rounds increases, the performance fluctuation range of the aggregated model with $\eta = 10^{-5}$ is smaller than that of $\eta = 10^{-4}$. In addition, we can also see from Figure 8 that when the learning rate is set too large, that is, $\eta = 10^{-2}$ and $\eta = 10^{-3}$, the performance of the aggregated model is difficult to improve with the increase of federated training rounds, and the fluctuation range is large. When the learning rate is set too small, that is, $\eta = 10^{-6}$, the performance of the aggregated model improves too slowly. Therefore, it is very important to select an appropriate learning rate for the model provider.

In the experiment of exploring the influence of the E on the aggregated model, we set the epoch E from 10 to 50 in turn, while other parameters, the training process and fusion process remain unchanged. Figure 8 shows the influence of different epochs on ACC and AUC values of the aggregated model.

From Figure 9, we can see that the performance of the aggregated model with different epochs can rise in a fluctuating state as the number of federated training rounds increases. The range of ACC is from to 0.38 to 0.80 and the range of AUC is from 0.67 to 0.95. When the epochs are set to be small, such as $E = 10$ or $E = 20$, the ACC and AUC values of the aggregated models rise slowly at the beginning, but with the increase of federated training rounds, the ACC and AUC values of the aggregated models can catch up with other aggregated models trained with other epochs. In addition, we can see that the ACC and AUC values of the aggregated models can achieve good performance after multiple federated training rounds whatever the epochs are. However, considering that the bigger epochs, the longer it takes to train a local model. Therefore, in practical application, we recommend that choosing an appropriate epoch value according to the actual situation, which can save the training time when the model can achieve the required performance.

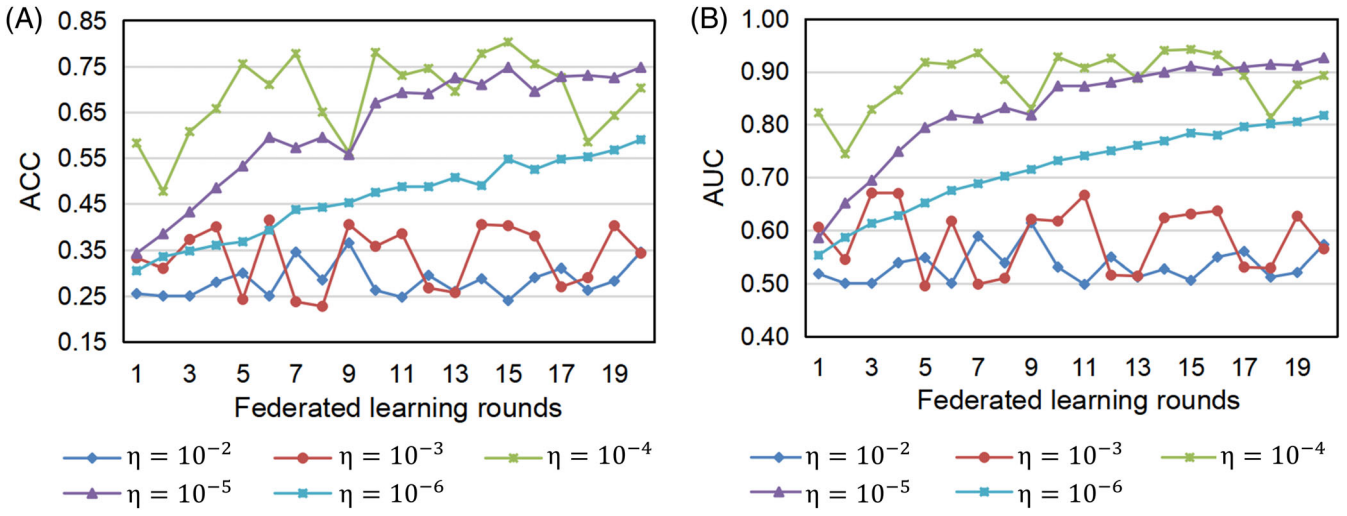

FIGURE 8 The influence of learning rates on the (A) Accuracy and (B) Area Under Receiver Operating Characteristic Curve values of the aggregated models [Color figure can be viewed at wileyonlinelibrary.com]

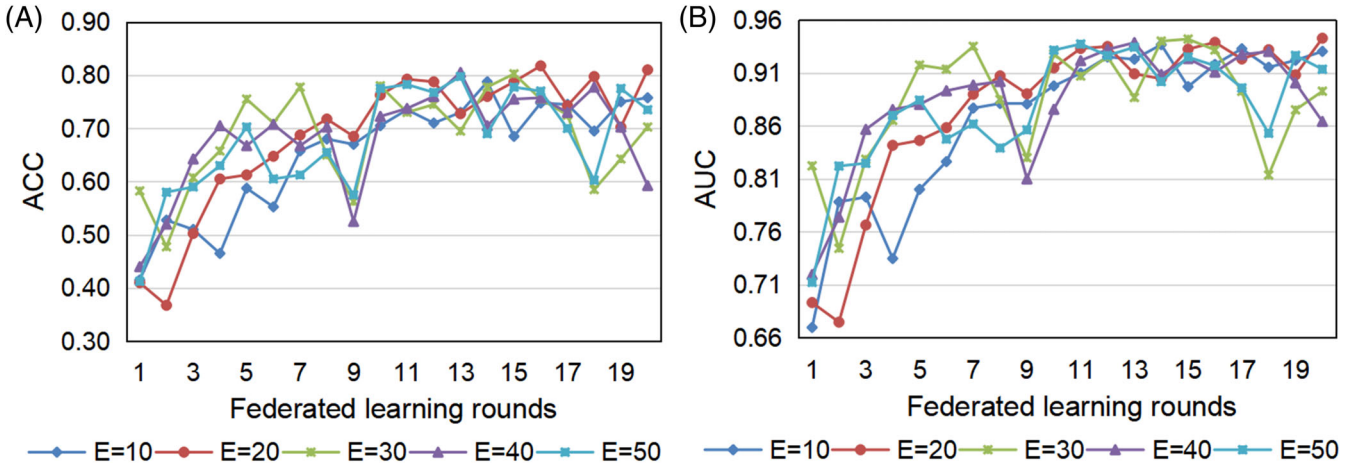

FIGURE 9 The influence of epochs on the (A) Accuracy and (B) Area Under Receiver Operating Characteristic Curve values of the aggregated models [Color figure can be viewed at wileyonlinelibrary.com]

From the above experiments, it can be seen that the learning rate has a great influence on the aggregated model, while the epoch has a small influence on the aggregated model. Both these two hyperparameters need to be set according to the situation in practical use.

5 | CONCLUSION

CPS and AI provide the ability to obtain and process large amounts of data from the physical world. If AI used in the MCPS to assist doctor diagnosis, medical resources can be used more efficiently and higher quality can be provided for medical process. In this paper, we propose a novel scheme based on federated learning for training disease diagnosis models from distributed medical image data in MCPS. The scheme is divided into three parts: the model provider, the server, and the consumer, and a detailed working process is designed for each part. This scheme can not only effectively solve the problems of privacy protection and model selection, but also ensure that consumers get a steadily improved disease diagnosis model automatically. We have carried out simulation experiments on a medical image dataset, and the experimental results prove the effectiveness and superiority of our proposed scheme. In future work, we will study the interpretability of the method used in our scheme to help improve our scheme and strengthen the trust relationship between consumers and our scheme.

ACKNOWLEDGEMENTS

This work is supported by the Natural Science Foundation of China (61672535, 61502540, 61977062), National Science Foundation of Hunan Province, China (2019JJ20025 and 2019JJ40406), the Earmarked fund for China Agriculture Research System, the 111 project under grant no. B18059 and the Fundamental Research Funds for the Central Universities of Central South University (2019zzts962). The authors declare that they have no conflict of interests.

ORCID

Kehua Guo https://orcid.org/0000-0003-4143-6399

REFERENCES

1. Cui J, Ren L, Wang X, et al. Pairwise comparison learning based bearing health quantitative modeling and its application in service life prediction. *Future Gener Comput Syst*. 2019;97:578-586.
2. Zhang J, Liu R, Yin K, et al. Intelligent collaborative localization among air-ground robots for industrial environment perception. *IEEE Trans Ind Electron*. 2019;66:9673-9681.
3. Gosman C, Cornea T, Dobre C, et al. Controlling and filtering users data in intelligent transportation system. *Future Gener Comput Syst*. 2018;78:807-816.

4. Rahman MA, Asyhari AT, Obaidat MS, et al. IoT-enabled light intensity-controlled seamless highway lighting system. *IEEE Syst J*. 2020;1-10.
5. Meng W, Li W, Wang Y, et al. Detecting insider attacks in medical cyber–physical networks based on behavioral profiling. *Future Gener Comput Syst*. 2020;108:1258–1266.
6. Frangi AF, Tsaftaris SA, Prince JL. Simulation and synthesis in medical imaging. *IEEE Trans Med Imaging*. 2018;37(3):673-679.
7. Lee I, Sokolsky O, Chen S, et al. Challenges and research directions in medical cyber–physical systems. *Proc IEEE*. 2012;100(1):75-90.
8. Long E, Lin H, Liu Z, et al. An artificial intelligence platform for the multihospital collaborative management of congenital cataracts. *Nature Biomed Eng*. 2017;1(2):1-8.
9. Guo K, Liu D, Li T, et al. MADP: an open and scalable medical auxiliary diagnosis platform. *Comput Sci Eng*. 2018;1-1.
10. Mcmahan HB, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*; 2016.
11. Litjens GJ, Kooi T, Bejnordi BE, et al. A survey on deep learning in medical image analysis. *Med Image Anal*. 2017;42:60-88.
12. Liu J, Zhang Z, Wong DW, et al. Automatic glaucoma diagnosis through medical imaging informatics. *J Am Med Inform Assoc*. 2013;20(6):1021-1027.
13. Havaei M, Davy A, Wardefarley D, et al. Brain tumor segmentation with deep neural networks. *Med Image Anal*. 2017;35:18-31.
14. Esteva A, Kuprel B, Novoa RA, et al. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*. 2017;542(7639):115-118.
15. Gibson E, Li W, Sudre CH, et al. NiftyNet: a deep-learning platform for medical imaging. *Comput Methods Programs Biomed*. 2018;158:113-122.
16. Yang Y, Zheng X, Guo W, et al. Privacy-preserving smart IoT-based healthcare big data storage and self-adaptive access control system. *Inform Sci*. 2019;479:567-592.
17. Dzwonkowski M, Papaj M, Rykaczewski R, et al. A new quaternion-based encryption method for DICOM images. *IEEE Trans Image Process*. 2015;24(11):4614-4622.
18. Heurix J, Fenz S, Rella A, et al. Recognition and pseudonymisation of medical records for secondary use. *Med Biol Eng Comput*. 2016;54(2–3):371-383.
19. Cong P, Zhou J, Wei T, et al. Personality-guided cloud pricing via reinforcement learning. *IEEE Trans Cloud Comput*. 2020. https://doi.org/10.1109/TCC.2020.2992461.
20. Konecný J, Mcmahan HB, Yu FX, et al. Federated learning: strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*; 2016.
21. Nishio T, Yonetani R. Client selection for federated learning with heterogeneous resources in mobile edge. *International Conference on Communications*. 2019;1-7.
22. Chen F, Dong Z, Li Z, et al. Federated meta-learning for recommendation. *arXiv preprint arXiv:1802.07876*; 2018.
23. Brisimi TS, Chen R, Mela T, et al. Federated learning of predictive models from federated electronic health records. *Int J Med Inform*. 2018;112:59-67.
24. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. *Comput Vis Pattern Recogn*. 2014.
25. Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions. *Comput Vis Pattern Recogn*. 2015;1-9.
26. He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. *Comput Vis Pattern Recogn*. 2016;770-778.
27. Bengio Y, Simard PY, Frasconi P, et al. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans Neural Netw*. 1994;5(2):157-166.
28. Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. Paper presented at: International Conference on Artificial Intelligence and Statistics, Sardinia, Italy; 2010:249–256.
29. Li L, Zhou J, Wei T, et al. Learning-based modeling and optimization for real-time system availability. *IEEE Trans Comput*. 2020. https://doi.org/10.1109/TC.2020.2991177.
30. Keuper J, Preundt F. Distributed training of deep neural networks: theoretical and practical limits of parallel scalability. Paper presented at: IEEE International Conference on High Performance Computing Data and Analytics, Hyderabad, India; 2016:19–26.
31. Jin PH, Yuan Q, Iandola F, et al. How to scale distributed deep learning. *arXiv preprint arXiv:1611.04581*; 2016.
32. Izmailov P, Podoprikhin D, Garipov T, et al. Averaging weights leads to wider optima and better generalization, Hyderabad, India; *arXiv preprint arXiv:1803.05407*; 2018.

How to cite this article: Guo K, Li N, Kang J, Zhang J. Towards efficient federated learning-based scheme in medical cyber-physical systems for distributed data. *Softw: Pract Exper*. 2020;1–16. https://doi.org/10.1002/spe.2894