SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

July 2024, Vol. 67, Iss. 7, 170306:1–170306:26
https://doi.org/10.1007/s11432-023-4011-4

Special Topic: XAI-Enabled Network Automation for 5G and Beyond: Move Closer to Zero-Touch

Adaptive 5G-and-beyond network-enabled interpretable federated learning enhanced by neuroevolution

Bin CAO[1,2*], Jianwei ZHAO[2], Xin LIU[3] & Yun LI[4]

[1]*State Key Laboratory of Reliability and Intelligence of Electrical Equipment, Hebei University of Technology, Tianjin* 300401*, China;*
[2]*School of Artificial Intelligence, Hebei University of Technology, Tianjin* 300401*, China;*
[3]*School of Economics and Management, Hebei University of Technology, Tianjin* 300401*, China;*
[4]*Industrial Artificial Intelligence Centre, Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China, Shenzhen* 518110*, China*

Received 15 August 2023/Revised 29 January 2024/Accepted 22 April 2024/Published online 27 June 2024

Abstract Mobile telemedicine systems based on the next-generation communication will significantly enhance deep fusion of network automation and federated learning (FL), but data privacy is a paramount issue in sectors like healthcare. This work hence considers FL augments 5G-and-beyond networks by training deep learning (DL) models without the need to exchange raw data. The substantial communication loads imposed on by extensive parameters involved in DL models are managed through adaptive scheduling mechanisms effectively. To address the opaque nature of DL models and to improve the interpretability of FL models, we introduce a convolutional fuzzy rough neural network specifically designed for medical image processing. We also develop a multiobjective memetic evolutionary algorithm to streamline and optimize the neural network architectures. Our comprehensive FL framework integrates smart scheduling, interpretable fuzzy rough logic, and neuroevolution. This framework is shown to improve communication efficiency, increase interpretability of diagnosis with protected privacy, and generate low-complexity neural architectures.

Keywords 5G-and-beyond network, interpretable federated learning, mobile telemedicine system, fuzzy rough set theory, neuroevolution

1 Introduction

Mobile telemedicine systems based on next-generation communication networks [1, 2] will promote the deep fusion of network automation and federated learning (FL), providing patients with more convenient mobile services, changing traditional mobile service models, and greatly expanding patients' treatment options. At present, mobile healthcare has developed into a new medical service model, playing an important role in mobile services. Mobile medical technology is widely used in health examinations, outpatient services, remote consultation, big data analysis, health management, and other fields. It has established a complete set of mobile medical processes to meet the needs of patients for fast, efficient, and convenient medical services.

Although mobile medical technology has gradually become more popular, there still exist some challenges, such as information security, privacy protection, medical expertise, and diagnostic decision-making accuracy. Among them, the challenges of data security and privacy protection are particularly prominent. Medical data are often stored across numerous dispersed institutions or individuals, and the system content within institutions is prone to unauthorized access, posing a great threat and impact to data security and personal privacy. Therefore, mobile healthcare needs to address the aforementioned issues from a technical perspective to provide more convenient, safe, and reliable medical services for patients and healthcare professionals. The advent of FL offers a reliable solution. FL can be used for multi-institutional cooperation, only transmitting the deep learning (DL) model but not patients' medical data [3]. Therefore, it can break through the data islands, attaining indirect medical data sharing. Owing to its privacy

* Corresponding author (email: caobin@scse.hebut.edu.cn)

protection characteristic for patients' medical image information, FL is attracting increasing attention and has been extensively applied to medical image analysis, such as ultrasound images [4], multisite 3D brain image [5], Chest X-ray images [6], and CT scan images [7].

DL has been widely used in medical imaging for automatic diagnosis. FL can benefit from DL models for accurate diagnosis and protect patient privacy information, especially in the medical imaging field. Usually, there are many parameters in a DL model, and the communication cost in FL is high if the local models are frequently uploaded to the central server. Accordingly, a novel dynamic fusion scheme was proposed to improve model performance and reduce the communication cost for COVID-19 detection in medical image data [8]. In [9], an asynchronous FL framework was presented to reduce the communication cost. Zhang et al. [10] proposed the communication-efficient federated continual learning method, which combines bidirectional compression and error compensation to simultaneously reduce communication cost and improve accuracy. Xu et al. [11] developed a highly–efficient FL framework, denoted by FedMax, which balances among clients with different computational capacities. The faster clients wait for the slower ones, and more workload can be assigned to the faster ones dynamically.

Convolutional neural networks (CNNs) have been successfully applied in the field of image recognition [12], especially for medical image processing [13], and the performance of CNNs is excellent. The key drawback is that deep CNNs are regarded as a black box, lacking interpretability [14]. Zhang et al. [15] designed a feature map loss term for the last few layers such that only one specific region in each feature map has high activation for a special target, so that the knowledge in the deeper convolutional layer is clearer to understand. The accuracy and interpretability requirements of CNN models in medical diagnosis are high [16]. Especially in the field of disease treatment and diagnosis using medical images, interpretable DL models have attracted wide interest. CNNs could be used to segment suspected regions of the images [17]. However, few studies have combined interpretability with FL for medical image processing.

Fuzzy theory emphasizes the fuzziness of knowledge in an information system and rough set theory focuses on the indiscernibility [18, 19]. Fuzzy rough neural networks (FRNNs) can be constructed and optimized via evolutionary algorithms (EAs) [20,21] to tackle complex problems in the real world [22]. Cao et al. [23] integrated the gene expression programming into an FRNN, which improved the effectiveness and explainability of the neural network. Yeganejou et al. [24] presented an interpretable model by combining a CNN and the interpretable fuzzy logic, where the CNN extracts features and then the fuzzy clustering proceeds with the processing.

Neuroevolution is able to automatically search the neural network architecture and hyperparameters [25], which can be implemented in the FL environment [26], enabling multiobjective optimization. In [27], the evolutionary FL model based on NSGANetV2 [28] not only achieved high medical image classification accuracy but also reduced network complexity. A CNN is an effective DL model [29], and neuroevolution can be used for designing DL architectures for COVID-19 diagnosis from X-ray images [30]. An evolutionary CNN with an attention mechanism was used for medical image segmentation, aiming at CT and MRI image datasets [31].

In this study, we address the communication efficiency, interpretability, and neuroevolution issues in FL. The contributions are outlined as follows.

(1) An adaptive 5G-and-beyond network-enabled communication scheduling scheme is proposed for FL. By uploading local model information based on their training statuses, the communication burden can be reduced.

(2) An interpretable convolutional FRNN (CFRNN) model is constructed. In the CNN supernet, based on the FRNN, the final expand layer is replaced by a fuzzification layer, fuzzy rules are formed by averaging, the absolute values of connection weights are regarded as rough membership degrees, the corresponding layer of which is considered the rough layer, and the following layer is the output layer, exhibiting clear interpretability.

(3) A flexible neural architecture search (NAS) space is designed, and to search light-weight neural architectures for the specific medical image classification task, a memetic multiobjective EA (MOEA) that focuses on optimizing the more complex validation accuracy objective while simultaneously considering the model complexity objective is proposed.

The rest of this article is organized in the following. The preliminaries for better understanding of this paper are in Section 2. The proposed adaptive 5G-and-beyond network-enabled FL framework is detailed in Section 3. Section 4 presents the experimental details. Finally, this paper is concluded in Section 5.

2 Preliminaries

2.1 Federated learning

FL was proposed by Google in 2016 [32], aiming at optimizing DL models based on distributed data without raw data sharing. Compared with centralized learning, FL is a decentralized data solution with various advantages [33]. First, the clients optimize local models with their own local data, and there is no data sharing among clients, protecting data privacy, which is crucial in the medical field [34]. Second, as the datasets are distributed in clients, the storage requirements and computational burden of each client are smaller [35]. Third, as clients can perform optimization in parallel, the time consumption can be reduced.

In an FL environment, there are a central server and multiple clients, which aggregates the uploaded models and train local models utilizing local data, respectively. Assuming there are K clients, the simple aggregation can be formulated as

$$\boldsymbol{w}^{\mathrm{g}} = \sum_{k=1}^{K} \frac{1}{K} \boldsymbol{w}_k^{\mathrm{l}}, \tag{1}$$

where $\boldsymbol{w}^{\mathrm{g}}$ denotes the aggregated model parameters, and $\boldsymbol{w}_k^{\mathrm{l}}$ denotes the model parameters of client k.

FL is a research hotspot in the fields of artificial intelligence and machine learning, and the related research mainly focuses on model performance, communication efficiency, privacy protection technology, application scenarios, and so on.

Improving model performance is a key concern of FL research. At present, many methods have been proposed, such as model compression, model fusion, model selection, and model evolution. Ge et al. [36] proposed a gradient quantization scheme based on clustering to achieve model compression and reduce communication cost. Zhou et al. [37] designed a spatio-temporal federated transfer learning framework for collaborative localization. Cheng et al. [38] proposed a federated transfer learning framework with client selection. To avoid choosing unreliable and low-quality clients, they used reinforcement learning for client selection to achieve high accuracy while saving costs. Zhu et al. [39] used a multiobjective evolutionary algorithm to optimize the architecture of a neural network model in FL by simultaneously minimizing communication costs and classification errors. A scalable network connectivity coding method was proposed to improve evolutionary efficiency.

In FL, there are multiple nodes, which cooperate with each other via communication. Therefore, communication efficiency is another key concern of FL research [40]. To date, researchers have proposed many methods to improve the communication efficiency of FL, such as adaptive communication, asynchronous communication, and compression communication. Tam et al. [41] proposed an adaptive communication scheme for edge FL for secure and real-time image classification in the Internet of Things. The scheme uses deep Q-learning algorithm to find the optimal resource control strategy. Zong et al. [42] introduced a parameter synchronization strategy based on the 2D-Ring network structure and a device placement algorithm based on 2D attention to minimize communication overhead. Wu et al. [43] used compressed sensing to compress and reconstruct local network models. To improve the accuracy, a genetic algorithm (GA) has been used to optimize the measurement matrix in compressed sensing. In addition, a method of staggered training and reconstruction has been proposed to improve the learning performance. Chen et al. [44] proposed the FedUC algorithm to select clients based on weight divergence, update increment, and loss to reduce communication costs. They also proposed a gradient compression and aggregation strategy based on dynamic weights.

In FL, privacy protection technology, which mainly includes encryption technology and differential privacy technology, is an important concern. Zhou et al. [45] proposed a network data analysis function architecture that supports FL based on partial homomorphic encryption to ensure model sharing and privacy protection. Li et al. [46] proposed an enhanced privacy protection mechanism based on generative adversarial networks and studied the privacy levels and convergence performance. FL can be applied to various scenarios, including finance [47], medical care [48], and transportation [49]. Therefore, studying the application scenarios of FL is another important direction.

As mentioned above, the study of FL involves various aspects; however, the study of FL + efficient communication + neuroevolution + FRNN requires further exploration.

2.2 Fuzzy logic

In fuzzy logic, an input is fuzzified to a membership degree via a membership function (MF):

$$f^{\mathrm{MF}}(u) \in [0,1], \tag{2}$$

where f^{MF} denotes the MF whose output is in $[0,1]$, and u denotes an input. Usually, the MF can take the form of a Gaussian function:

$$f^{\mathrm{MF}}(u) = \mathrm{e}^{-\frac{(u-c)^2}{2\sigma^2}}, \tag{3}$$

where c and σ are parameters.

Based on MFs, a fuzzy rule can be constructed as

$$\text{IF } f_1^{\mathrm{MF}}(u_1) \wedge f_2^{\mathrm{MF}}(u_2) \wedge \cdots \wedge f_{N^{\mathrm{In}}}^{\mathrm{MF}}(u_{N^{\mathrm{In}}}), \text{ THEN } O^{\mathrm{Rule}}, \tag{4}$$

where $\wedge$ denotes the logic operator AND and can be replaced by the product operator $\times$, N^{In} denotes the number of inputs, $f_i^{\mathrm{MF}}, i \in \left\{1,2,\ldots,N^{\mathrm{In}}\right\}$ denotes the MF for input i, $u_i, i \in \left\{1,2,\ldots,N^{\mathrm{In}}\right\}$ denotes input i, and O^{Rule} denotes the output of the fuzzy rule, which can be detailed as

$$O^{\mathrm{Rule}} = a_0 + \sum_{i=1}^{N^{\mathrm{In}}} a_i u_i, \tag{5}$$

where a denotes the parameter.

Through defuzzification, the final output can be generated:

$$O = \frac{\sum_{n=1}^{N^{\mathrm{Rule}}} p_n^{\mathrm{Rule}} O_n^{\mathrm{Rule}}}{\sum_{n=1}^{N^{\mathrm{Rule}}} p_n^{\mathrm{Rule}}}, \tag{6}$$

where O denotes the final output value, N^{Rule} denotes the fuzzy rule number, and p_n^{Rule} denotes the probability of fuzzy rule n, detailed as

$$p_n^{\mathrm{Rule}} = \prod_{i=1}^{N^{\mathrm{In}}} f_i^{\mathrm{MF}}(u_i). \tag{7}$$

FL is a kind of distributed machine learning paradigm, fuzzy logic is a kind of data processing and decision technology, and the two have both differences and similarities. The research on the combination of the two is still in the development stage, and relevant studies can be summarized as follows:

(1) Federated fuzzy learning: The fuzzy system is also an important machine learning technology. Further combined with FL, it can be applied to extended scopes and scenarios, learn distributed data, and play a more important role. Related research includes federated fuzzy clustering and federated fuzzy neural networks. Hu et al. [50] proposed federated multi-view fuzzy C-means clustering. Zhang et al. [51] proposed a federated fuzzy neural network (FNN) for tackling non-independent and identically distributed (non-IID) problems and data uncertainty. This method maintains a set of global rules and personalized subsets for the client in the server and updates the rules via evolutionary learning.

(2) Fuzzy selection and fusion: In the FL environment, there are many clients involved in learning, and the data generated and used by different clients are different. Thus, how to select and fuse them is an important research topic. Vinita et al. [52] proposed a vehicle selection technique based on fuzzy logic to select suitable vehicles in the roadside unit and base station as clients to participate in FL training. Yoo et al. [53] proposed a fuzzy clustered FL algorithm in which each local generator can belong to multiple clusters, helping to generate high-precision models using different features while solving the problem of insufficient training data.

Combining FL and fuzzy logic is an interesting topic, and it is also considered in this study.

2.3 Rough set theory

Without introducing any external knowledge, samples in the dataset can be partitioned based on their features. The dataset can be divided into several non-overlapping subsets with respect to each feature. However, the subsets with respect to different features may overlap. After analyzing all features, lots of subsets can be formed, constituting the knowledge base. Therefore, a target set can be approximated by the knowledge base:

$$\underline{R}(A) = \cup\left\{s | s \subseteq A, s \in \left\{\emptyset, B^{\text{Rough}}, \cup, \cap\right\}\right\}, \tag{8}$$

$$\overline{R}(A) = \cup\left\{s | s \wedge A \neq \emptyset, s \in \left\{\emptyset, B^{\text{Rough}}, \cup, \cap\right\}\right\}, \tag{9}$$

where A denotes a target set, $\underline{R}$ ($\overline{R}$) denotes the lower (upper) approximation, which constitutes the rough set $R(A) = \left\langle \underline{R}(A), \overline{R}(A) \right\rangle$, B^{Rough} denotes the knowledge base, $\cup$ ($\cap$) denotes the union (intersection) of sets, and $\left\{\emptyset, B^{\text{Rough}}, \cup, \cap\right\}$ denotes all definable sets derived from the knowledge base as well as the union and intersection operations on them.

The rough MF can be formulated as

$$f_A^{\text{RMF}}(s) = \frac{\|s \wedge A\|}{\|s\|}, \tag{10}$$

where f^{RMF} denotes the rough MF, and $\|s\|$ denotes the cardinality of set s.

Rough set theory is a practical method to deal with uncertain data. Its combination with FL is a research direction worth exploring. At present, the research on the combination of FL and rough set theory can be summarized in the following aspects:

(1) Federated rough learning: FL and rough learning can be combined to extract knowledge from dispersed data. Some scholars have proposed federated rough systems. Przybyła-kasperek et al. [54] combined FL with rough set theory for the first time, defined the global reducts from the local reducts generated by decentralized data, and evaluated the local reducts with local units.

(2) Model fusion: In FL, it is necessary to conduct collaborative training for distributed clients, and how to integrate dispersed models is an important problem. Rough set theory can be applied to the fusion of different models. Hu et al. [55] proposed a rough selection-based strategy to populate unbalanced or empty classes for clients in FL.

(3) Interpretability: Utilizing rough set theory, interpretable knowledge can be summarized from the data. Combined with FL, some researchers have studied interpretable FL systems. Pedrycz et al. [56] investigated the combination of FL with information granules including the rough set, built and improved information granules based on collaborative clustering, and studied interpretability and FL based on functional rules.

In general, the combination of FL and rough set theory is a worthy research topic to explore. The distributed learning paradigm of FL helps to apply rough set theory to more scenarios.

2.4 Fuzzy rough set and fuzzy rough neural network

Assuming there are several fuzzy sets, $\{F_1, F_2, \ldots, F_{N^{\text{FS}}}\}$, as $f^{\text{MF}} \in [0,1]$, the following holds:

$$\sup_u \min\left\{f_{F_i}^{\text{MF}}(u), f_{F_j}^{\text{MF}}(u)\right\} \leqslant 1, \ \forall i, j \in \left\{1, 2, \ldots, N^{\text{FS}}\right\}, \tag{11}$$

where $f_{F_i}^{\text{MF}}(u)$ denotes the MF of input u to fuzzy set F_i, and N^{FS} denotes the number of fuzzy sets.

Then, the fuzzy sets can be utilized to describe the target sets, $\{D_1, D_2, \ldots, D_{N^{\text{TS}}}\}$, as follows [57]:

$$f_{\underline{D_i}}^{\text{MF}}(F_j) = \inf_{u \in D_i} \max\left\{1 - f_{F_j}^{\text{MF}}(u), f_{D_i}^{\text{MF}}(u)\right\}, \tag{12}$$

$$f_{\overline{D_i}}^{\text{MF}}(F_j) = \sup_{u \in D_i} \min\left\{f_{F_j}^{\text{MF}}(u), f_{D_i}^{\text{MF}}(u)\right\}, \tag{13}$$

where $\underline{D_i}$ ($\overline{D_i}$) denotes the lower (upper) approximation of the fuzzy rough set $\left\langle \underline{D_i}, \overline{D_i} \right\rangle$, and $f_{D_i}^{\text{MF}}(u) = \{0,1\}$ denotes the MF of input u to target set D_i.

Algorithm 1 Neuroevolution

Input: Population size: NP, generation number: N_{GEN}.
Output: Final population: $S_{\mathrm{final}}^{\mathrm{pop}}$.
1: Initialize population S_0^{pop} with size NP;
2: **for** $g \in \{1, 2, \ldots, N_{\mathrm{GEN}}\}$ **do**
3: Select parents from S_{g-1}^{pop};
4: Generate offspring based on the selected parents via crossover and mutation;
5: Combine S_{g-1}^{pop} and offspring to generate S_g^{pop};
6: **end for**
7: $S_{\mathrm{final}}^{\mathrm{pop}} = S_{N_{\mathrm{GEN}}}^{\mathrm{pop}}$.

The fuzzy rough MF can be formulated as

$$f_{D_i}^{\mathrm{FRMF}}(u) = \frac{1}{\sum_{j=1}^{N^{\mathrm{FS}}} f_{F_j}^{\mathrm{MF}}(u)} \sum_{j=1}^{N^{\mathrm{FS}}} f_{F_j}^{\mathrm{MF}}(u) f_{D_i}^{\mathrm{RMF}}(F_j), \tag{14}$$

where $f_{D_i}^{\mathrm{FRMF}}(u)$ denotes the fuzzy rough MF of input u to target set D_i. If $\sum_{j=1}^{N^{\mathrm{FS}}} f_{F_j}^{\mathrm{MF}}(u) = 0$, $f_{D_i}^{\mathrm{FRMF}}(u) = 0$.

By combining the neural network (NN) and fuzzy rough set theory, FRNNs [19, 22, 58] can be constructed. Based on the fuzzy logic in Subsection 2.2, there are six layers in an FRNN:

(1) L_1: Input layer. Each node in this layer simply copies each input feature value.

(2) L_2: Fuzzification layer. Each input node in the previous layer connects to several fuzzification nodes such that each input feature value is transformed to several membership degrees via fuzzification according to (2) and (3).

(3) L_3: Fuzzy rule layer. As in (4), each fuzzy rule node connects multiple fuzzification nodes in the previous layer, and its output membership degrees determine the fire strength of the fuzzy rule node, but there is no O^{Rule}.

(4) L_4: Rough layer. Different from the FNN, there is an additional rough layer immediately after the fuzzy rule layer. The positive weights connecting the fuzzy rule nodes and rough nodes represent rough membership degrees.

(5) L_5: Consequence layer. For each rough node, there exists a consequence node, formulated as (5).

(6) L_6: Output layer. As in (6), based on the fuzzy rough membership degree outputs of the rough nodes and outputs of the consequence nodes, we can obtain the final output via defuzzification.

Fuzzy rough set, NNs, and FL are all machine learning techniques, and they can be combined in various ways. Their effective combinations can develop a series of data processing methods. Some scholars have made some explorations about this.

(1) Fuzzy rough set: Both fuzzy and rough sets are useful mathematical tools for characterizing ambiguous and partial data, and they work well together [59]. Qi et al. [60] affirmed the important role of fuzzy rough set in the process of data mining and proposed two kinds of fuzzy covering-based rough set models.

(2) FRNNs: Fuzzy rough set theory can be combined with NNs to construct FRNNs. Zhao et al. [61] constructed FRNN architectures for mobile sink scheduling in wireless sensor networks and proposed distributed parallel multiobjective neural evolutionary algorithms for optimization.

(3) Federated FRNNs: FRNNs can be combined with FL to learn from distributed data and applied to a wider range of scenarios. Liu et al. [27] combined the CNN with fuzzy rough set theory to construct a convolutional interval type-2 fuzzy rough FL model, and searched high-precision and low-complexity network architectures based on multiobjective NAS. Liu et al. [62, 63] further proposed three deep fuzzy rough convolutional NN architectures for large-scale multiobjective federated neuroevolution.

Overall, the combination of fuzzy rough set, NNs, and FL is an interesting and promising research topic. However, the existing relevant research is still very limited, and further exploration is needed.

2.5 Neuroevolution

Neuroevolution refers to optimizing hyperparameters of NNs via applying evolutionary ideas. Its procedure is detailed in Algorithm 1.

Neuroevolution utilizes EAs for optimizing NNs, avoiding some disadvantages of using gradient descent, such as requiring gradients and falling into local optima. Neuroevolution mimics the biological evolution

to learn building blocks, block connections, hyperparameters and even the optimization algorithms, and it possesses inherent global exploration and parallelism characteristics [64].

In an FL environment, multiple clients learn from local data based on machine learning models such as DL. By combining FL with neuroevolution, neuroevolutionary learning will be based on distributed data, which is both a challenge and opportunity for traditional neuroevolutionary algorithms [65]. Some scholars have explored this aspect, summarized as follows:

(1) CNNs: Zhu et al. [66] proposed a real-time federated neuroevolution algorithm by randomly sampling multiple clients to train randomly sampled subnets, which simultaneously optimizes model performance and reduces costs of calculation and communication.

(2) FNNs: Zhang et al. [51] studied a federated fuzzy neural network and proposed an evolutionary rule learning strategy, which evolves global rules to generate new local rules.

(3) Fuzzy rough CNNs: Liu et al. [27] combined FL with neuroevolution to explore architectures of convolutional interval type-2 FRNNs for medical security by simultaneously considering model accuracy and complexity. Then, they further studied this issue and designed three kinds of deep fuzzy rough CNN architectures [62].

To sum up, in the distributed environment of FL, some scholars have conducted neuroevolution studies based on different NNs, such as CNNs, FNNs, and fuzzy rough CNNs. However, there are few studies combining FL, neuroevolution, fuzzy set theory, rough set theory, and NNs, which need more efforts of researchers.

3 Adaptive 5G-and-beyond network-enabled interpretable federated learning

In this section, by using the classification task as an example, we introduce details of the adaptive communication in FL for the 5G-and-beyond network, denoted by FedAdap. Then, we propose an interpretable CFRNN with neuroevolution.

3.1 Adaptive communication

The procedure of the FL based on the proposed FedAdap scheme is detailed in Algorithm 2 and illustrated in Figure 1. To reduce the communication burden of uploading and downloading DL model parameters in the FL, a straightforward way is to exchange information at every predefined number of epochs instead of every epoch. To further improve efficiency, all clients upload their training states to the server (the solid yellow arrow in Figure 1), and the server adaptively decides whether clients upload the model information for aggregation. Three types of training information are considered:

(1) The accuracy improvement count (AIC). After each epoch of training, with respect to every client $k \in \{1, 2, \ldots, K\}$ (K denotes the amount of clients), if the training accuracy shows improvement, the AIC $N^k_{\rm imp}$ will be updated (the line 14 of Algorithm 2). The AIC will also be compared to the AIC threshold $N^{\rm th}_{\rm imp}$, and if $N^k_{\rm imp} \geqslant N^{\rm th}_{\rm imp}$ as well as $i\%N^{\rm th}_{\rm epoch} = 0$ (i denotes the current epoch number), the client k will be labeled as a candidate client (the solid red arrow in Figure 1 and the line 20 of Algorithm 2).

(2) The accuracy improvement degree (AID). Let $v^k_{\rm ba}$ denote the best historical accuracy, $v^k_{\rm cba}$ denote the best current accuracy, $v^{\rm idl}_{\rm acc}$ denote the ideal accuracy, and $r^{\rm th}_{\rm imp}$ denote the accuracy ratio. If $v^k_{\rm cba} - v^k_{\rm ba} \geqslant r^{\rm th}_{\rm imp}\left(v^{\rm idl}_{\rm acc} - v^k_{\rm ba}\right)$ as well as $i\%N^{\rm th}_{\rm epoch} = 0$ (i denotes the current epoch number), the client k will be labeled as a candidate client (the solid red arrow in Figure 1 and the line 20 of Algorithm 2).

(3) The stagnation count (SC). Let $N^k_{\rm stag}$ denote the SC, $N^{\rm th}_{\rm stag}$ denote the SC threshold, and $v^{\rm th}_{\rm stag}$ denote the stagnation threshold. If the accuracy of the i-th training epoch does not improve by more than $v^{\rm th}_{\rm stag}$ compared to $v^k_{\rm cba}$ as well as $v^k_{\rm cba} < v^{\rm idl}_{\rm acc}$, $N^k_{\rm stag}$ will be updated (the line 13 of Algorithm 2). If $N^k_{\rm stag} \geqslant N^{\rm th}_{\rm stag}$ as well as $i\%N^{\rm th}_{\rm epoch} = 0$ (i denotes the current epoch number), the client k will be labeled as a candidate client (the solid red arrow in Figure 1 and the line 20 of Algorithm 2).

Finally, if more than half of the clients become candidates, the central server signals them to upload their local model parameters (the dashed green arrow in Figure 1 and the line 25 of Algorithm 2), and sends back the aggregated model information (the dashed purple arrow in Figure 1 and the line 28 of Algorithm 2).

Algorithm 2 Adaptive federated learning

Input: Epoch number: N_{epoch}, epoch threshold: $N_{\text{epoch}}^{\text{th}}$, improvement count threshold: $N_{\text{imp}}^{\text{th}}$, stagnation count threshold: $N_{\text{stag}}^{\text{th}}$, stagnation value threshold: $v_{\text{stag}}^{\text{th}}$, accuracy ratio threshold: $r_{\text{imp}}^{\text{th}}$, ideal accuracy: $v_{\text{acc}}^{\text{idl}} = 1.0$.
Output: Final model weight: $\boldsymbol{w}_{\text{final}}^{\text{g}}$.

The server builds the DL model;
The server initializes the model weights $\boldsymbol{w}^{\text{g}}$;
The server scatters the model to clients;
for $k \in \{1, 2, \ldots, K\}$ **do**
    $N_{\text{comm}}^{k} = 0$;
    $v_{\text{ba}}^{k} = 0$;
    $v_{\text{cba}}^{k} = 0$;
    $N_{\text{imp}}^{k} = 0$;
    $N_{\text{stag}}^{k} = 0$;
end for
for $i \in \{1, 2, \ldots, N_{\text{epoch}}\}$ **do**
    Clients train the local models;
    Update N_{stag}^{k} if the improvement is not above $v_{\text{stag}}^{\text{th}}$ and the ideal accuracy has not reached;
    Update v_{cba}^{k} and N_{imp}^{k} if the training accuracy is improved;
    if $i \% N_{\text{epoch}}^{\text{th}} = 0$ **then**
        if $i = N_{\text{epoch}}$ **then**
            All clients are candidates for model uploading and updating;
        else
            if $N_{\text{imp}}^{k} \geqslant N_{\text{imp}}^{\text{th}}$ or $N_{\text{stag}}^{k} \geqslant N_{\text{stag}}^{\text{th}}$ or $v_{\text{cba}}^{k} - v_{\text{ba}}^{k} \geqslant r_{\text{imp}}^{\text{th}}(v_{\text{acc}}^{\text{idl}} - v_{\text{ba}}^{k})$ **then**
                Client k becomes a candidate for updating;
            end if
        end if
        if The number of candidates is above $\frac{K}{2}$ **then**
            Update v_{ba}^{k}, N_{imp}^{k}, and N_{comm}^{k};
            Candidate clients upload model weights;
            The server validates the received model weights;
            The server aggregates the model weights based on the training data sizes;
            Scatter the updated $\boldsymbol{w}^{\text{g}}$ to the candidate clients;
        end if
    end if
end for
$\boldsymbol{w}_{\text{final}}^{\text{g}} = \boldsymbol{w}^{\text{g}}$.

3.2 Convolutional fuzzy rough neural network

The backbone CNN is the modified MobileNetV3 with 2 blocks similar to that in [27], which is suitable for mobile clinics. In Figure 2, the two-block MobileNetV3 and the proposed CFRNN are illustrated. By comparing these two architectures, we can see that in CFRNN, the final expand layer is replaced by the fuzzification layer, and the following layers can be regarded as the fuzzy rule layer, the rough layer, and the output layer. Their differences can be summarized as follows:

(1) In the original final expand layer, the input feature maps are expanded to more channels via a series of 1×1 filters,

$$O_{c,h,w}^{\text{FEL}} = \sigma^{\text{FEL}}\left(\sum_{i=1}^{c_{\text{In}}^{\text{FEL}}} W_{c,i}^{\text{FEL}} I_{i,h,w}^{\text{FEL}}\right), \tag{15}$$
$$\text{s.t.} \begin{cases} c = 1, 2, \ldots, c_{\text{Out}}^{\text{FEL}}, \\ h = 1, 2, \ldots, h_{\text{In}}^{\text{FEL}}, \\ w = 1, 2, \ldots, w_{\text{In}}^{\text{FEL}}, \end{cases}$$

where O^{FEL} (I^{FEL}) denotes the output (input) of the final expand layer, σ^{FEL} denotes the activation function called HSWISH, c denotes the index of the channel, h (w) denotes the ordinal of the height (width) of the feature map, $c_{\text{In}}^{\text{FEL}}$ denotes the channel number of input feature maps of the final expand layer, W^{FEL} denotes the filter weights, $c_{\text{Out}}^{\text{FEL}}$ denotes the channel number of output feature maps of the final expand layer, and $h_{\text{In}}^{\text{FEL}}$ ($w_{\text{In}}^{\text{FEL}}$) denotes the height (width) of input feature maps.

In the fuzzification layer, two Gaussian MFs, as in 3, are utilized to transform each feature value to two membership degrees; thus, the number of channels is doubled:

$$O^{\text{FZ}} = \left[\underline{O}^{\text{FZ}}, \overline{O}^{\text{FZ}}\right], \tag{16}$$

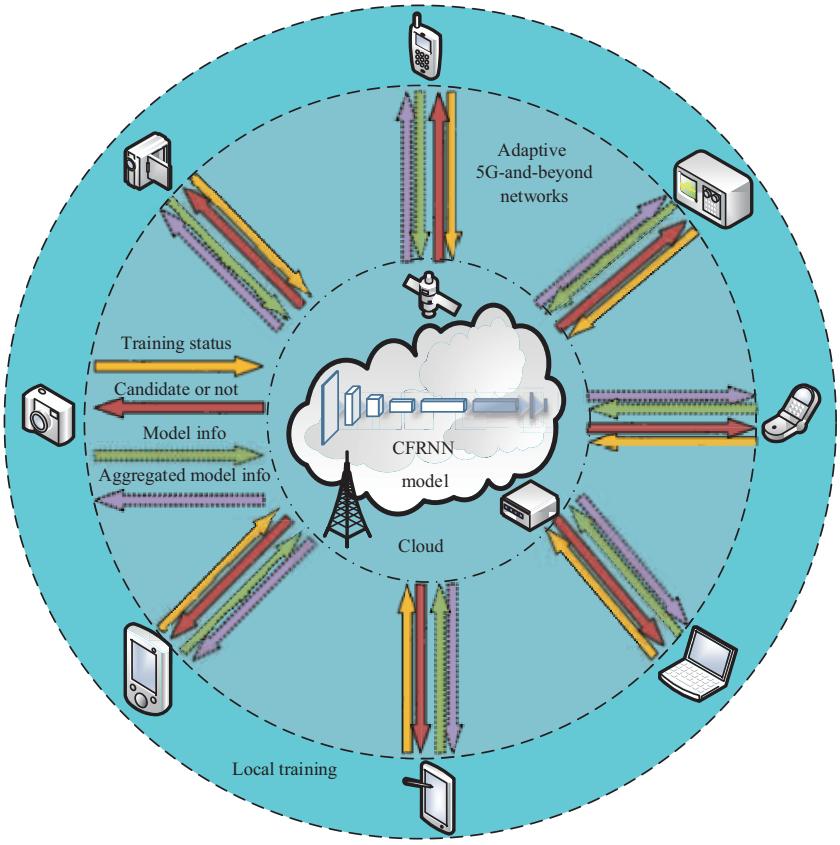

Figure 1 (Color online) FedAdap scheme for the 5G-and-beyond network with FL.

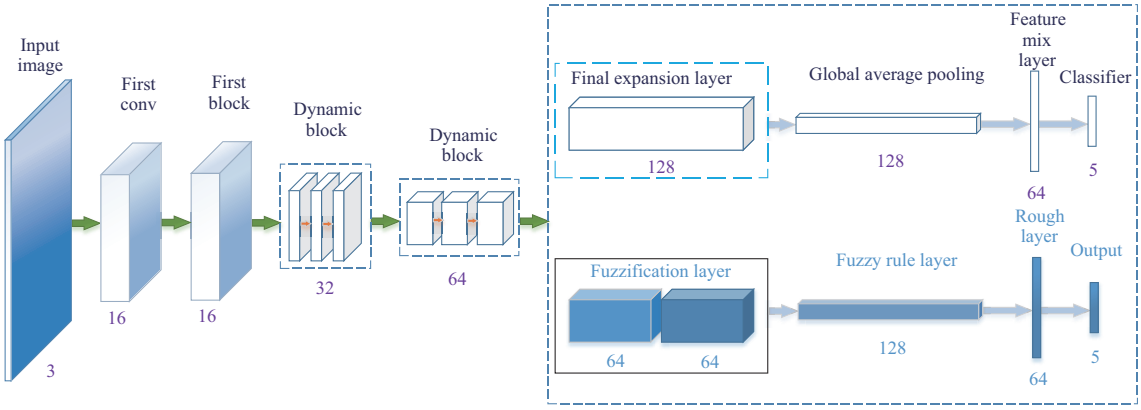

Figure 2 (Color online) Two-block MobileNetV3 and the CFRNN structures.

$$\underline{O}_{i,h,w}^{\mathrm{FZ}} = \underline{f}^{\mathrm{mb}}\left(I_{i,h,w}^{\mathrm{FZ}}\right), \tag{17}$$

$$\text{s.t.} \begin{cases} i = 1, 2, \ldots, c_{\mathrm{In}}^{\mathrm{FZ}}, \\ h = 1, 2, \ldots, h_{\mathrm{In}}^{\mathrm{FZ}}, \\ w = 1, 2, \ldots, w_{\mathrm{In}}^{\mathrm{FZ}}, \end{cases}$$

$$\overline{O}_{i,h,w}^{\mathrm{FZ}} = \overline{f}^{\mathrm{mb}}\left(I_{i,h,w}^{\mathrm{FZ}}\right), \tag{18}$$

$$\underline{f}^{\mathrm{mb}}(u) = \mathrm{e}^{\frac{-(u-c_1)^2}{2\sigma_1^2}}, \tag{19}$$

$$\overline{f}^{\mathrm{mb}}(u) = \mathrm{e}^{\frac{-(u-c_2)^2}{2\sigma_2^2}}, \tag{20}$$

where O^{FZ} denotes the output of the fuzzification layer, $\underline{O}^{\mathrm{FZ}}$ and $\overline{O}^{\mathrm{FZ}}$ denote two groups of membership degree maps, $c_{\mathrm{In}}^{\mathrm{FZ}}$ denotes the channel number of input feature maps of the fuzzification layer, $h_{\mathrm{In}}^{\mathrm{FZ}}$ ($w_{\mathrm{In}}^{\mathrm{FZ}}$) denotes the height (width) of input feature maps, and $\underline{f}^{\mathrm{mb}}$ and $\overline{f}^{\mathrm{mb}}$ denote two MFs, which have parameters of c_1, σ_1, c_2, and σ_2. In Figure 2, there are 64 input feature maps ($c_{\mathrm{In}}^{\mathrm{FZ}} = 64$), and through fuzzification, two groups of membership degree maps ($\underline{O}^{\mathrm{FZ}}$ and $\overline{O}^{\mathrm{FZ}}$) can be obtained, each of which has 64 channels. Therefore, the number of output channels is $c_{\mathrm{Out}}^{\mathrm{FZ}} = 128$. By replacing the final expand layer with the fuzzification layer, the number of parameters is decreased from $c_{\mathrm{Out}}^{\mathrm{FZ}} \times c_{\mathrm{In}}^{\mathrm{FZ}} = 128 \times 64$ to 2×2 (i.e., the number of parameters in 19 and 20), significantly reducing the complexity.

(2) In the two-block MobileNetV3, the global pooling is utilized to reduce each feature map to a single value, which can be regarded as the fuzzy rule layer in the CFRNN. In this way, the fire strength of a fuzzy rule can be obtained by averaging the membership degrees in the same feature map:

$$I^{\mathrm{FL}} = O^{\mathrm{FZ}}, \tag{21}$$

$$O_c^{\mathrm{FL}} = \frac{\sum_{h=1}^{h_{\mathrm{In}}^{\mathrm{FL}}} \sum_{w=1}^{w_{\mathrm{In}}^{\mathrm{FL}}} I_{c,h,w}^{\mathrm{FL}}}{h_{\mathrm{In}}^{\mathrm{FL}} w_{\mathrm{In}}^{\mathrm{FL}}}, \quad \text{s.t. } c = 1, 2, \ldots, c_{\mathrm{In}}^{\mathrm{FL}}, \tag{22}$$

where I^{FL} and O^{FL} denote the input and output of the fuzzy rule layer, respectively, $c_{\mathrm{In}}^{\mathrm{FL}}$ denotes the input channel number, and $h_{\mathrm{In}}^{\mathrm{FL}}$ ($w_{\mathrm{In}}^{\mathrm{FL}}$) denotes the height (width) of input membership degree maps.

(3) The feature mix layer utilizes multiple 1×1 kernels to mix the input features. In the CFRNN, this layer can be regarded as the rough layer, and the kernel weights are rough membership degrees. Because the rough membership degrees are in $[0, 1]$, the absolute values of the kernel weights are used without adjustment. The details of this layer are as follows:

$$I^{\mathrm{RL}} = O^{\mathrm{FL}}, \tag{23}$$

$$O_c^{\mathrm{RL}} = \sum_{i=1}^{c_{\mathrm{In}}^{\mathrm{RL}}} \mathrm{fabs}\left(W_{c,i}^{\mathrm{RL}}\right) I_i^{\mathrm{RL}}, \quad \text{s.t. } c = 1, 2, \ldots, c_{\mathrm{Out}}^{\mathrm{RL}}, \tag{24}$$

where I^{RL} and O^{RL} denote the input and output of the rough layer, respectively, $c_{\mathrm{Out}}^{\mathrm{RL}}$ denotes the output channel number, $c_{\mathrm{In}}^{\mathrm{RL}}$ denotes the input channel number, fabs denotes the absolute value function, and $\mathrm{fabs}(W^{\mathrm{RL}})$ denotes the rough membership degrees.

(4) Finally, the classifier layer is a full-connection layer with the number of output nodes equal to the number of classes for classification. In the CFRNN, this is the output layer, and similar to the rough layer, the absolute connection weights are utilized. Moreover, the bias is removed. For simplicity, different from the FRNN, there is no consequence layer, or the outputs of consequence nodes are all fixed to 1. The details of this layer are as follows:

$$I^{\mathrm{OUT}} = O^{\mathrm{RL}}, \tag{25}$$

$$O_c^{\mathrm{OUT}} = \sum_{i=1}^{c_{\mathrm{In}}^{\mathrm{OUT}}} \mathrm{fabs}\left(W_{c,i}^{\mathrm{OUT}}\right) I_i^{\mathrm{OUT}}, \quad \text{s.t. } c = 1, 2, \ldots, c_{\mathrm{Out}}^{\mathrm{OUT}}, \tag{26}$$

where I^{OUT} and O^{OUT} denote the input and output of the output layer, respectively, $c_{\mathrm{Out}}^{\mathrm{OUT}}$ denotes the output class number, $c_{\mathrm{In}}^{\mathrm{OUT}}$ denotes the input channel number, and W^{OUT} denotes the weights of the output layer.

In the fuzzification layer, the Gaussian MFs are utilized to transform the input feature maps. Because the Gaussian MFs are quite different from the operations in the convolution and full-connection layers, the traditional initialization methods are inappropriate. Therefore, a manually designed initialization scheme is proposed:

(1) Parameter c: As the batch normalization is applied in the prior layer, the values in the input feature maps are approximately in $[-1, 1]$. To associate high membership values to pixels with high

Table 1 Hyperparameters in the search space

Symbol	Description	Range
L_{kn}	Kernel size	$\{3, 5, 7\}$
R_{xp}	Expansion rate	$\{1, 2, 3, 4\}$
D_{b}	Depth of each block	$\{1, 2, 3, 4\}$
S_{in}	Input image size	$\{128, 132, 136, \ldots, 256\}$
T_{act}	Activation function type	{relu, relu6, tanh, sigmoid, h_swish, h_sigmoid}
W_{bs}	Base stage width	$\{8, 16\}$ $\{8, 24, 32\}$ $\{16, 24, 32, 48, 64\}$ $\{8, 16, 32, 64\}$

absolute activation, the c_1 and c_2 parameters in the two Gaussian MFs are initialized to -1 and 1, respectively.

(2) Parameter σ: As the input feature map values are in $[-1, 1]$, in order to keep the output value neither too large nor too small to facilitate training, the parameters σ_1 and σ_2 are simply initialized to 1.

(3) Otherwise, if c and σ are randomly assigned values in $[-1, 1]$ or a similar range, the MFs will assign high membership degrees for random values in $[-1, 1]$, and a small σ will lead to small output values. This will decrease the interpretability and greatly reduce the convergence rate, and the network could become trapped in a local optimum with poor accuracy.

3.3 Neuroevolution

Based on the proposed CFRNN supernet, a flexible search space is designed:

(1) The activation functions in the dynamic blocks are optimized, and there are six types.

(2) The base stage widths between blocks are treated as variables and optimized, corresponding to the numbers of channels of the first convolution and the first block, two dynamic blocks as well as the rough layer in Figure 2.

All hyperparameters considered in the search space are listed in Table 1.

For the optimization objectives, the classification accuracy on the validation dataset and FLOPs are simultaneously considered. In the neuroevolution, the network accuracy is usually more important than the complexity and more difficult to optimize. However, in the MOEAs, different objectives are treated equally. Therefore, in the MOEA-based neuroevolution, the objectives of validation accuracy and complexity are equally optimized without emphasis. For this issue, based on NSGANetV2 [28], a memetic algorithm is proposed by combining an MOEA and an EA, which are NSGA-II [67] and GA, respectively. The MOEA simultaneously optimizes both objectives, and the EA focuses on improving the accuracy. Corresponding to the MOEA and EA, there are two archives, one of which stores the nondominated solutions and the other records the solutions with better accuracy. The pseudocodes are detailed in Algorithm 3.

Algorithm 3 Federated memetic NSGA-II for Neuroevolution

Input: Initial population size: P_0, evolution population size: P, population portion of GA: p, iteration number: N_{ITER}, generation number: N_{GEN}, number of selected individuals: N_{real}.
Output: Final population: $S_{\mathrm{arc}}^{\mathrm{pop}}$.
Train the supernet based on FedAdap in the FL environment;
Initialize population $S_{\mathrm{arc}}^{\mathrm{pop}}$ with size P_0;
Sample subnets from the supernet, train subnets based on FedAdap in the FL environment, and train the surrogate model;
for $i \in \{1, 2, \ldots, N_{\mathrm{ITER}}\}$ **do**
  Initialize the NSGA-II archive $S_{\mathrm{M}}^{\mathrm{arc}}$ and the GA archive $S_{\mathrm{S}}^{\mathrm{arc}}$ based on the nondominated solutions in $S_{\mathrm{arc}}^{\mathrm{pop}}$;
  for $g \in \{1, 2, \ldots, N_{\mathrm{GEN}}\}$ **do**
    Generate $(1 - p) \times P$ offspring via NSGA-II based on $S_{\mathrm{M}}^{\mathrm{arc}}$;
    Generate $p \times P$ offspring via GA based on $S_{\mathrm{S}}^{\mathrm{arc}}$;
    Obtain the combined offspring population S_g^{off}, and predict the accuracy based on the surrogate;
    Combine $S_{\mathrm{M}}^{\mathrm{arc}}$ and S_g^{off} and generate the updated $S_{\mathrm{M}}^{\mathrm{arc}}$ via nondominated sorting and crowding distance;
    Combine $S_{\mathrm{S}}^{\mathrm{arc}}$ and S_g^{off} and generate the updated $S_{\mathrm{S}}^{\mathrm{arc}}$ based on accuracy;
  end for
  $S_{\mathrm{evo}}^{\mathrm{pop}} = S_{\mathrm{M}}^{\mathrm{arc}} \bigcup S_{\mathrm{S}}^{\mathrm{arc}}$;
  Select N_{real} individuals from $S_{\mathrm{evo}}^{\mathrm{pop}}$, sample subnets from the supernet, train subnets based on FedAdap in FL environment, obtain the real accuracy, update $S_{\mathrm{arc}}^{\mathrm{pop}}$, and update the surrogate model;
end for

4 Experimentation

4.1 Data description

The utilized medical image dataset is the Lung and Colon Cancer Histopathological Images [68], denoted by LC25000. LC25000 is an augmented dataset from Health Insurance Portability and Accountability Act (HIPAA) compliant and validated sources, which comprise 750 images evenly distributed in 5 classes. There are 25000 images in LC25000, each of which includes 768×768 pixels and belongs to one of five classes: Lung benign tissue, Lung adenocarcinoma, Lung squamous cell carcinoma, Colon adenocarcinoma, and Colon benign tissue. Each class has 5000 images. The dataset is divided into three parts for training, validation, and testing with sizes of 15000, 5000, and 5000, respectively.

4.2 Experimental setting

In the considered FL scenario, there are 5 clients. Accordingly, the training set is divided into five parts, and two schemes are designed.

(1) Balanced data size: Each client has the same number of training images.

(2) Unbalanced data size (UBDS): The numbers of training images in clients are unbalanced, which are 1000, 2000, 3000, 4000, and 5000, respectively.

For the backpropagation training, we employ the stochastic gradient descent with an initial learning rate of 0.05, cosine learning rate scheduling, momentum of 0.9, and Nesterov.

To validate the proposed FedAdap, FedAvg [32], FedUC [44], and FedPC [69] have been compared. The default FL communication scheme of FedAvg [32] is that every $N_{\text{epoch}}^{\text{th}}$ epochs, all clients upload their local model information to the central server, and the central server updates the global model via aggregation before scattering back the updated global model information. Based on the default scheme, every $N_{\text{epoch}}^{\text{th}}$ epochs, FedAdap checks whether clients update their local model information according to Algorithm 2. Chen et al. [44] proposed FedUC, in which priority values are calculated based on weight divergence with the global model, weight increment with the local historical model, and training loss. Then, clients are selected for aggregation. In this study, we have reproduced this client selection scheme in FedUC (denoted as FedUCsc) and conducted a comparative experiment. In terms of parameters, clients upload their training state every $N_{\text{epoch}}^{\text{th}}$ training epochs as in Table 2, and the server then calculates the priority value and selects the client for aggregation. During each selection, a certain number of clients are selected, which is 10 out of 50 in [44]. In this study, we set it to 3, that is, 3 out of 5 clients are selected. After weighted model aggregation based on the sample sizes in clients, the server distributes the model information to all clients. Cao et al. [69] proposed FedPC, in which a best client is selected to upload its model to the server based on the local dataset size and the training loss. For other clients, only ternary vectors are calculated and uploaded. Finally, the updated global model will be scattered to all clients. In this study, we reproduced FedPC, and every $N_{\text{epoch}}^{\text{th}}$ training epochs, clients upload their model or ternary vectors.

The parameter settings are listed in Table 2. The first seven parameters are utilized in Algorithm 2. There are two values for each of N_{epoch}, $N_{\text{epoch}}^{\text{th}}$, $N_{\text{imp}}^{\text{th}}$, and $N_{\text{stag}}^{\text{th}}$. The first value is for supernet training, and the second is for subnet training in neuroevolution. Parameters of P_0, P, N_{real}, N_{ITER}, and N_{GEN} are utilized in Algorithm 3.

4.3 Results and analysis

4.3.1 *Federated supernet learning*

Based on Table 1, two extreme architectures are trained: the minimum-complexity network and the supernet with maximum complexity. Figure 3 illustrates the communication times. For the minimum-complexity architecture (Figures 3(a) and (b)), the maximum communication counts of clients are 11 and 12 for the balanced and unbalanced training set partitions, respectively. On average, clients in FedAdap communicate 9.6 times for balanced data partition. Contrarily, we can easily compute that each client following the default static FL of FedAvg [32] communicates $\frac{N_{\text{epoch}}}{N_{\text{epoch}}^{\text{th}}} = \frac{200}{10} = 20$ times. Therefore, the proposed FedAdap reduces the communication burden by 52%. For the unbalanced data partition, the communication counts vary from 10 to 12, and the average is 11, which is 55% of that of FedAvg. For the supernet (Figures 3(c) and (d)), the communication counts of all clients are 13 for the balanced data

Table 2 Parameter settings of adaptive communication of FL and multiobjective federated neuroevolution

Symbol	Description	Value
N_{epoch}	Epoch number	200 or 5
$N_{\text{epoch}}^{\text{th}}$	Epoch threshold	10 or 1
$N_{\text{imp}}^{\text{th}}$	Improvement count threshold	10 or 2
$N_{\text{stag}}^{\text{th}}$	Stagnation count threshold	20 or 2
$v_{\text{stag}}^{\text{th}}$	Stagnation value threshold	0.001%
$r_{\text{imp}}^{\text{th}}$	Accuracy ratio threshold	0.1
$v_{\text{acc}}^{\text{idl}}$	Ideal accuracy	1.0
P_0	Initial population size	100
P	Evolution population size	40
N_{real}	Number of selected individuals	8
N_{ITER}	Number of iterations	30
N_{GEN}	Number of generations	20

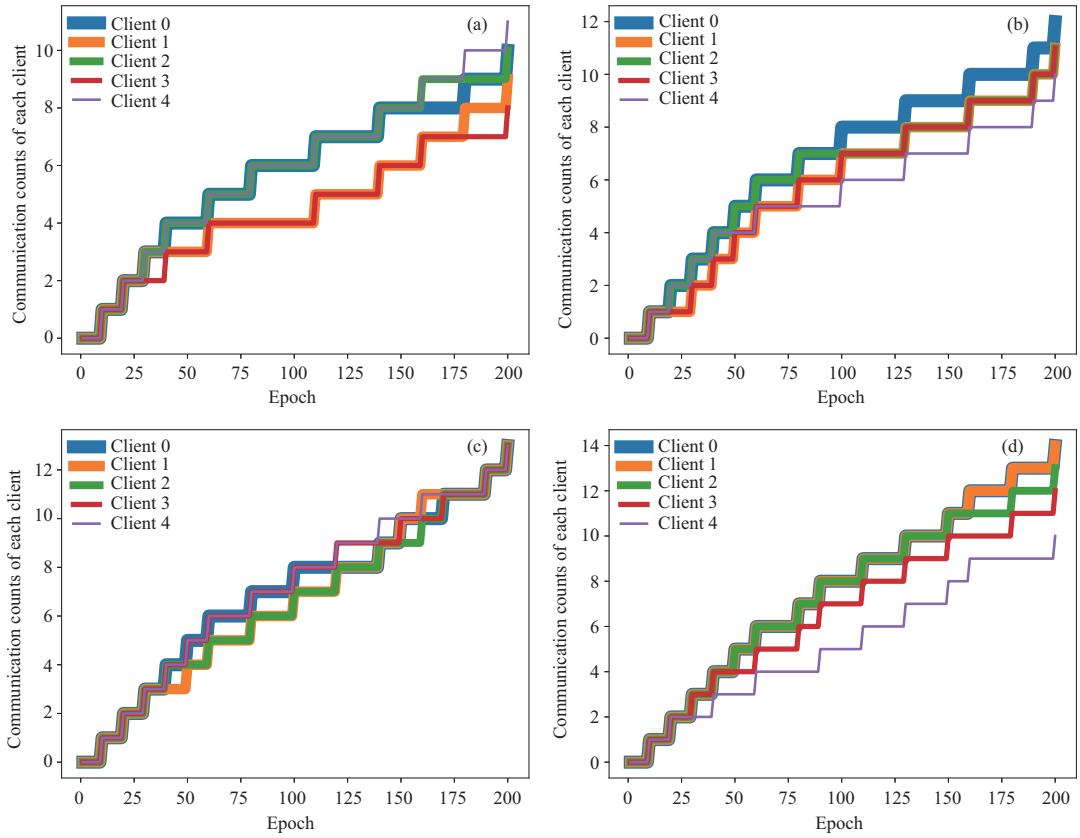

Figure 3 (Color online) Communication times. (a) MIN-CFRNN-FedAdap; (b) MIN-CFRNN-FedAdap-UBDS; (c) MAX-CFRNN-FedAdap; (d) MAX-CFRNN-FedAdap-UBDS.

partition. For the unbalanced data partition, the range and average of communication counts of clients are $[10.0, 14.0]$ and 12.6, respectively. The communication burden is reduced to 65% and 63% for the two data partition schemes compared to the static FL of FedAvg.

In terms of communication cost, after each evaluation of training states in FedUCsc, a fixed number of selected clients will upload model information, so the communication volume is fixed. As illustrated in Figure 4, 3 clients are selected every $N_{\text{epoch}}^{\text{th}}$ epochs, and different clients are selected during the training. The aggregated model will be distributed afterwards. The frequency of communication in FedUCsc is the same as that of FedAvg, and obviously more than that of the proposed FedAdap. Due to the gradient compression strategy, the communication load in FedUCsc is reduced to some extent.

For FedPC, as illustrated in Figure 5, one client is selected to upload its model and others upload ternary vectors. Then, the updated global model will be scattered to all clients. Therefore, the communication

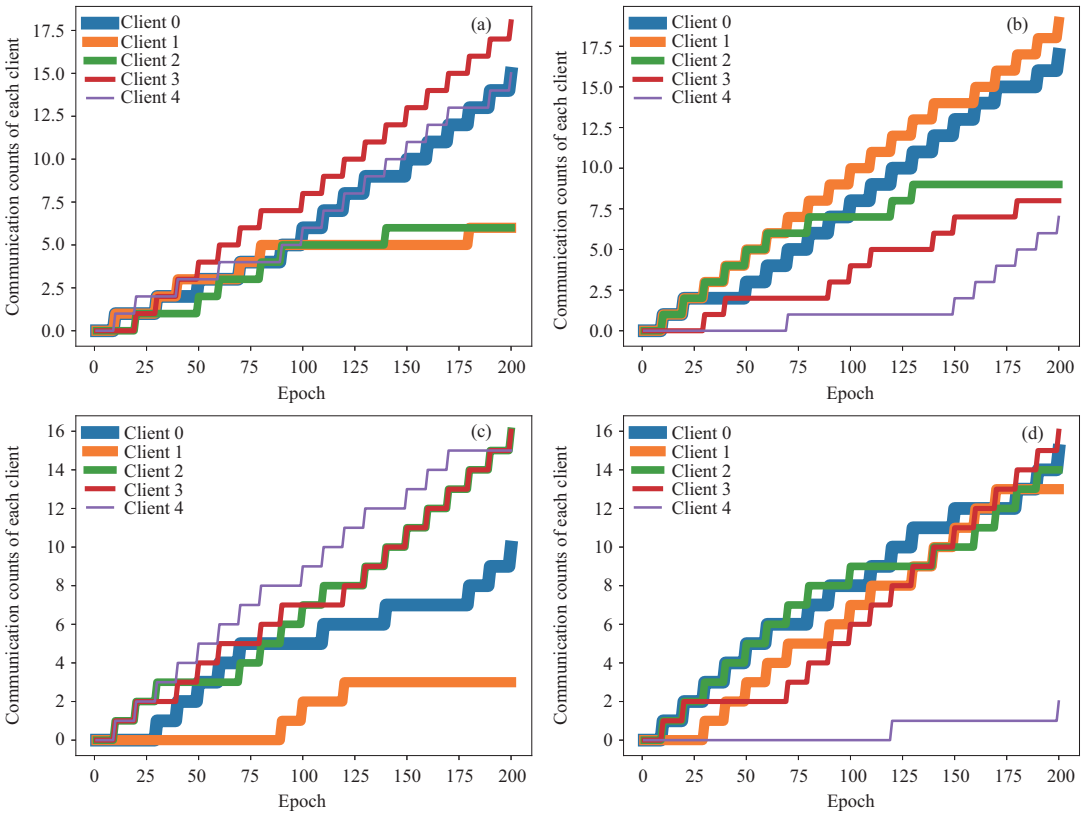

Figure 4 (Color online) Communication times of the selection scheme in FedUCsc. (a) MIN-CFRNN-FedUCsc; (b) MIN-CFRNN-FedUCsc-UBDS; (c) MAX-CFRNN-FedUCsc; (d) MAX-CFRNN-FedUCsc-UBDS.

burden of FedPC is much higher than that of FedAdap.

The accuracy curves on the training and validation sets are illustrated in Figure 6. We can observe that for the models trained with FedAvg, FedAdap, and FedUCsc, between two adjacent communication checks, the validation accuracies remain the same. The reason is that, for the training accuracies, they are averaged for all clients after each epoch, while the validation accuracy is only updated when checking the training states every $N_{\text{epoch}}^{\text{th}} = 10$ training epochs. The accuracy of the minimum-complexity network is much inferior to that of the supernet, which is also confirmed in Table 3. In Figures 6(a) and (b), we can see that the CFRNN in the FL environment has poor accuracies on the training set and validation set, which are significantly inferior to those of the model not trained in FL environment. In Figure 6(c), the training accuracies of all curves approximate to 100%. In Figure 6(d), there is little difference among the models in different experimental settings during the training process, but there are differences in the final accuracies. Therefore, in the proposed FedAdap and the compared FedUCsc environment, the performance loss of the trained model is small for supernets. It can be seen from Figures 6(b) and (d) that the performance of the model on the verification set fluctuates more significantly than that on the training set (Figures 6(a) and (c)). In addition, this fluctuation is more obvious in the simple network and FL environment. In FL environment, the performance of the model on the verification set does not easily converge stably, especially in the early and middle stages of training. Meanwhile, this phenomenon is mild for the supernet. For example, in Figure 6(b), the accuracy rate of the minimum-complexity network model still fluctuates greatly in the middle and late stages; however, in Figure 6(d), although the model fluctuates slightly in the middle stage, it converges stably in the later stage. In Figures 6(b) and (d), valid-FR-FedAdap-UBDS shows the most obvious fluctuation.

For clarity, the accuracy values are listed in Tables 3 and 4, where values in parentheses denote validation accuracies or epoch numbers.

(1) When training in the FL environment, training, verification, and test accuracies are slightly reduced, especially for architectures with the minimum complexity.

(2) In non-FL environments, for the minimum-complexity network, CNN and CFRNN without FL show optimal training accuracies lower than 96%, higher validation accuracies but lower than 98%, and

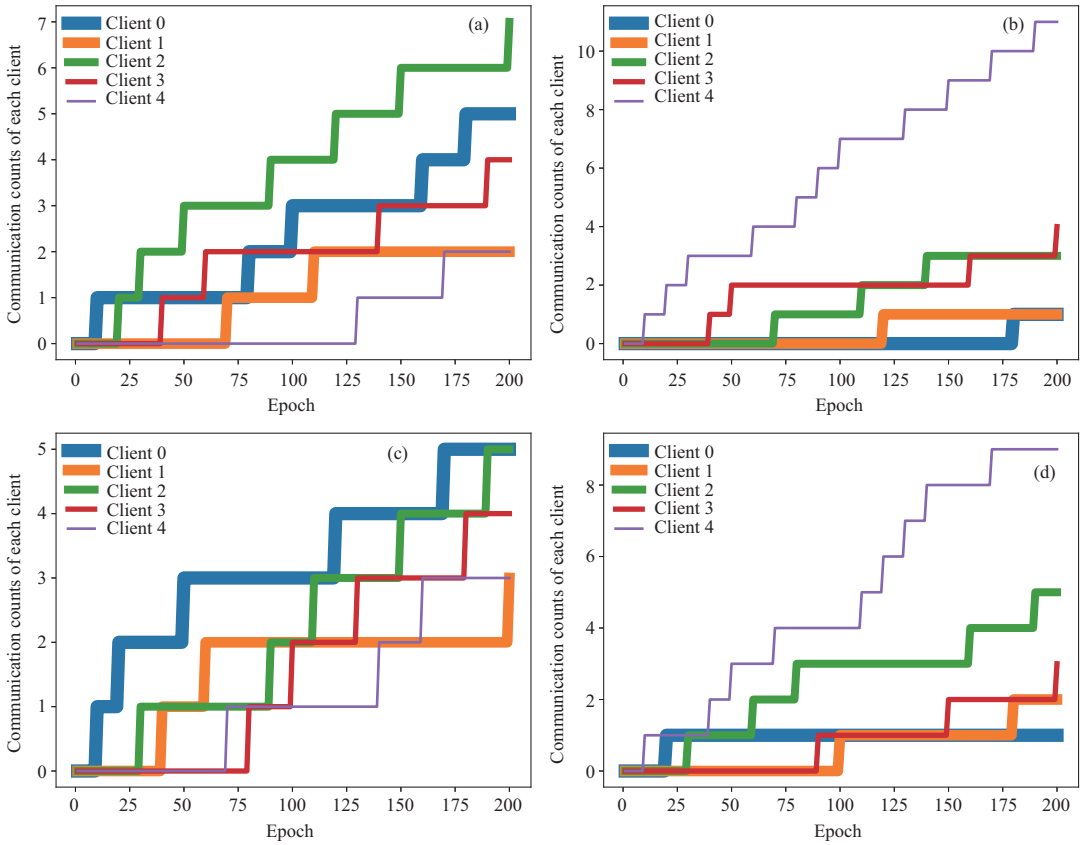

Figure 5 (Color online) Communication times of the selection scheme in FedPC. (a) MIN-CFRNN-FedPC; (b) MIN-CFRNN-FedPC-UBDS; (c) MAX-CFRNN-FedPC; (d) MAX-CFRNN-FedPC-UBDS.

test accuracies slightly higher than validation accuracies.

(3) In non-FL environments, CNN and CFRNN supernets get training accuracies higher than 99%, and validation and test accuracies are both 100%. From the number of epochs, it can be seen that for the training set, the convergence speeds of the supernet and the minimum-complexity network have a very small difference, but for the verification set, the convergence speed of the supernet is significantly faster than that of the minimum-complexity network.

(4) For minimum-complexity network in the FL environment, training, validation, and testing accuracies in the FedAdap environment are slightly better than those of FedAvg when balanced data partitioning is used. Those of FedUCsc are better than those of FedAvg. Compared to FedUCsc, FedPC is only worse in validation accuracy. FedPC is better than FedAvg, FedAdap, and FedUCsc. For unbalanced data partition, the training accuracy of FedAdap is the same as that in FedAvg, but the verification and test accuracies are slightly higher than in FedAvg. However, the training, verification, and test accuracies are slightly lower than when balanced data partition is used in the FedAdap environment. The performance of FedUCsc is better than that of FedAvg. Comparing FedUCsc and FedAdap, FedUCsc is better than FedAdap for the training and validation accuracies but worse for the test accuracy. FedPC is better than FedAvg, FedAdap, and FedUCsc, except that its validation accuracy is worse than that of FedUCsc.

(5) For supernets in the FL environment, when the data partition is balanced, the validation accuracy of the model trained in the FedAdap environment is lower than that in the FedAvg environment, but the training and test accuracies are slightly higher. FedUCsc is better than FedAdap and FedAvg for the test accuracy, better than FedAvg and worse than FedAdap for the training accuracy, but worse than FedAvg and better than FedAdap for the validation accuracy. FedPC is only worse than FedAvg and FedAdap for the training accuracy and only better than FedUCsc for the validation accuracy. For unbalanced data partition, the training, validation, and test accuracies of models in the FedAdap, FedUCsc, and FedPC environments are lower than those in the FedAvg environment. FedUCsc is better than FedAdap for the training accuracy, but worse than them for the validation and test accuracies. FedPC is worse than FedAdap and FedUCsc, except that its test accuracy is the same as that of FedUCsc.

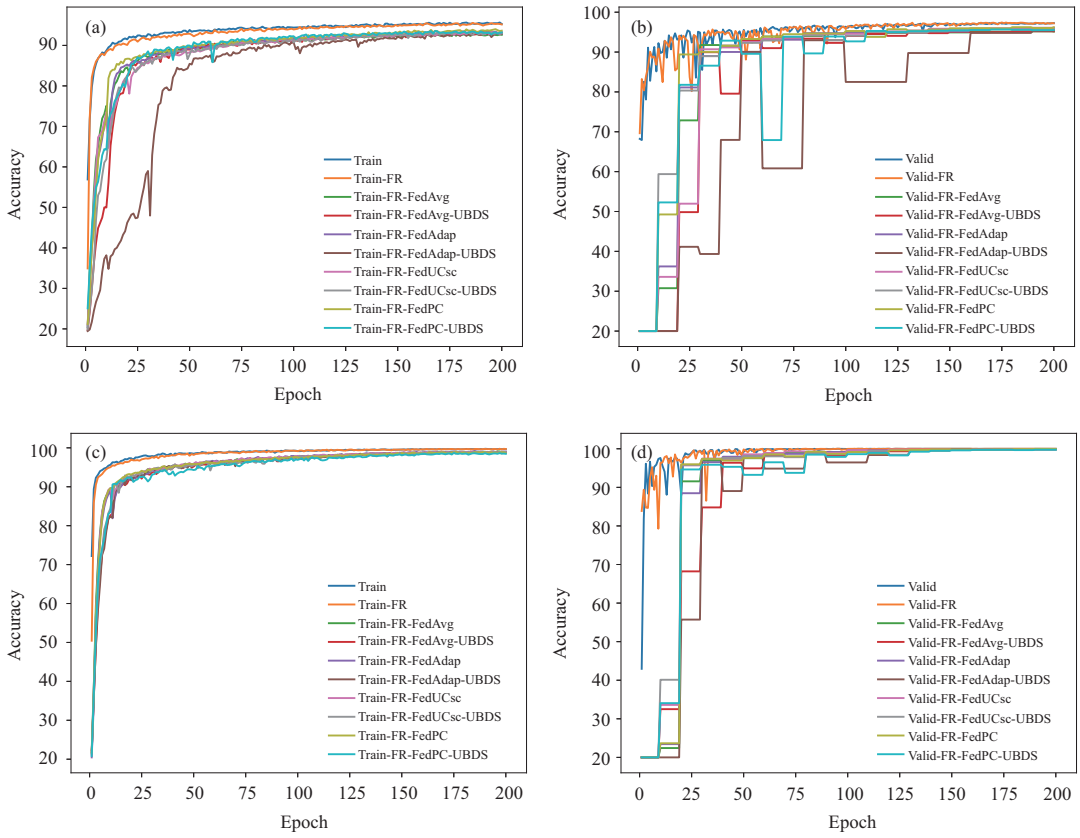

Figure 6 (Color online) Training and validation accuracy curves. 'FR' denotes CFRNN. (a) MIN training; (b) MIN validation; (c) MAX training; (d) MAX validation.

Table 3 Best training and validation accuracies of the minimum- and maximum-complexity models and corresponding epoch numbers as well as final test accuracies[a)]

		CNN	CFRNN	CFRNN-FedAvg	CFRNN-FedAvg-UBDS	CFRNN-FedAdap	CFRNN-FedAdap-UBDS
Minimum complexity	Best acc	95.66 (97.32)	95.57 (97.42)	93.00 (95.20)	93.22 (95.22)	93.56 (95.84)	93.22 (95.40)
	Epoch #	198 (173)	185 (174)	176 (189)	177 (189)	185 (199)	186 (199)
	Test acc	97.76	97.86	95.78	95.62	96.30	95.82
Maximum complexity	Best acc	99.76 (100.0)	99.76 (100.0)	99.07 (99.92)	99.04 (99.96)	99.21 (99.88)	98.91 (99.88)
	Epoch #	189 (92)	196 (58)	195 (179)	182 (169)	184 (189)	178 (159)
	Test acc	100.0	100.0	99.80	99.98	99.86	99.90

a) 'CNN' denotes the two-block MobileNetV3. Values in brackets denote validation accuracies or epoch numbers.

Table 4 Best training and validation accuracies of the minimum- and maximum-complexity models and corresponding epoch numbers as well as final test accuracies with respect to FedUCsc and FedPC

		CFRNN-FedUCsc	CFRNN-FedUCsc-UBDS	CFRNN-FedPC	CFRNN-FedPC-UBDS
Minimum complexity	Best acc	93.35 (96.14)	93.43 (95.74)	94.01 (96.24)	93.59 (95.66)
	Epoch #	186 (199)	181 (169)	196 (179)	167 (169)
	Test acc	95.98	95.78	96.84	96.34
Maximum complexity	Best acc	99.10 (99.90)	99.00 (99.80)	98.95 (99.98)	98.69 (99.76)
	Epoch #	193 (179)	198 (199)	175 (189)	174 (189)
	Test acc	99.96	99.78	99.94	99.78

(6) The accuracy gaps with respect to the training, verification, and test sets of the supernet models without FL and with FL are smaller than 1.07%, 0.24%, and 0.22%, respectively. The difference between the accuracies of the model trained in FedAvg, FedAdap, FecUCsc, and FedPC is small.

4.3.2 *Discussion on initialization of model parameters*

Table 5 lists the accuracies of the model after randomly initializing the fuzzification layer parameters. The values in the table correspond to those in Table 3. $x\downarrow$ and $x\uparrow$ respectively indicate that compared

Table 5 Model performance after randomly initializing parameters of the fuzzification layer[a)]

		CNN	CFRNN	CFRNN-FedAvg	CFRNN-FedAdap	CFRNN-FedAvg-UBDS	CFRNN-FedAdap-UBDS
Minimum complexity	Best acc	–	0.22 ↓ (0.44 ↓)	0.06 ↓ (0.38 ↑)	1.12 ↓ (1.14 ↓)	0.30 ↓ (0.04 ↓)	0.17 ↑ (0.08 ↑)
	Epoch #	–	6 ↑ (9 ↓)	15 ↑ (50 ↓)	7 ↓ (50 ↓)	10 ↑ (30 ↓)	12 ↑ (30 ↓)
	Test acc	–	0.20 ↓	0.04 ↓	1.36 ↓	0.20 ↑	0.30 ↑
Maximum complexity	Best acc	–	0.09 ↓ (0.00 =)	0.88 ↓ (0.52 ↓)	0.46 ↓ (0.32 ↓)	2.51 ↓ (2.04 ↓)	0.54 ↓ (0.36 ↓)
	Epoch #	–	0 = (28 ↑)	4 ↓ (0 =)	10 ↓ (10 ↑)	15 ↑ (0 =)	6 ↑ (40 ↑)
	Test acc	–	0.00 =	0.22 ↓	0.02 ↓	1.44 ↓	0.32 ↓

a) $x \downarrow$ and $x \uparrow$ denote x lower and higher than the corresponding value in Table 3; 0 = denotes values are equivalent.

with the corresponding values in Table 3, the accuracy rate decreases and increases by x after random initialization of the fuzzy layer parameters. "=" indicates that random initialization is no different than initialization with a specific value.

In Table 5, most of the values show that the performance of the model is reduced by different degrees after random initialization. The corresponding parameter change curve of the randomly initialized fuzzification layer is shown in Figure 7. Table 6 shows the final parameter values.

For the minimum-complexity network, MIN-CFRNN-FedAdap has the largest relative test accuracy reduction (1.36 ↓). As can be seen in Table 6, the parameters of the two MFs are relatively similar. As can be seen from Figure 8(c), the values of parameter c in the two MFs almost overlap during the entire training process, and although the values of parameter σ are positive and negative, the absolute values are relatively consistent. This situation is also found during the training of MIN-CFRNN. Although there is little difference in the final MFs in MIN-CFRNN-FedAvg-UBDS, the parameters are out of step during training. This may result in a decrease in training accuracy, but a slight increase in validation and test accuracy. As can be seen from Figures 8(b) and (e), there is a large difference between the two MFs. Corresponding to the values in Table 5, it can be seen that MIN-CFRNN-FedAvg only has a small decrease in the accuracies of training and testing, which is almost negligible, and the verification accuracy is improved. For MIN-CFRNN-FedAdap-UBDS, the accuracies of training, verification, and testing are improved. For the number of epochs in Table 5, there is no clear pattern to follow.

For the supernet, it can be seen from Table 5 that all accuracy values have decreased. Among them, MAX-CFRNN-FedAvg-UBDS shows the most obvious decrease. Corresponding to Figure 8(i), it can be seen that the parameters of the two MFs are relatively consistent throughout the training process. As can be seen from Table 6, the corresponding values of parameter c are almost equal, but the absolute values of parameter σ are slightly different. In Table 5, MAX-CFRNN is the one with the smallest differences. As can be seen in Table 6, the two MFs are very different, and the parameter change curve in Figure 8(f) is very stable and almost symmetric about the axis. The initialization parameter values are very similar to those of the proposed initialization scheme. The values of two MFs' parameter c are positive and negative, and the absolute values are similar. The values of parameter σ are also positive and negative, and the absolute values are similar. Therefore, the proposed initialization scheme is helpful to better train the model.

Figure 8 shows the changing curves of parameters during training after using the proposed special initialization scheme. It can be seen that the values of parameter c of the two MFs are always positive and negative during the whole training process, and the values of parameter σ may be consistent or slightly different. For the architecture with the minimum complexity, the changing curves of parameters fluctuate greatly, and the values of the two parameters are significantly different. For the supernet, except for MAX-CFRNN-FedAvg, the changing curves of parameter σ of other models almost coincide during the whole training process. In Figures 8(f), (h), (i), and (j), the values of parameter c of the two MFs are positive and negative, with little difference in absolute values, and the values of parameter σ are almost the same. In Figure 8(g), the values of parameter c of the two MFs are quite different, and the values of parameter σ are also different. By comparing MAX-CNN and MAX-CFRNN in Table 3, it can be seen that there is no difference in the accuracies of training, verification, and testing of the two models, and only the convergence speed on the verification set differs. Therefore, the characteristics of the trained MFs may be different, but it does not affect the performance of the model.

4.3.3 *Comparison with existing work*

Compared to the results in [27], the differences can be summarized as follows:

(1) The frequency of model aggregation is reduced from every epoch to every $N_{\text{epoch}}^{\text{th}}$ epochs. Moreover, the FedAdap is proposed.

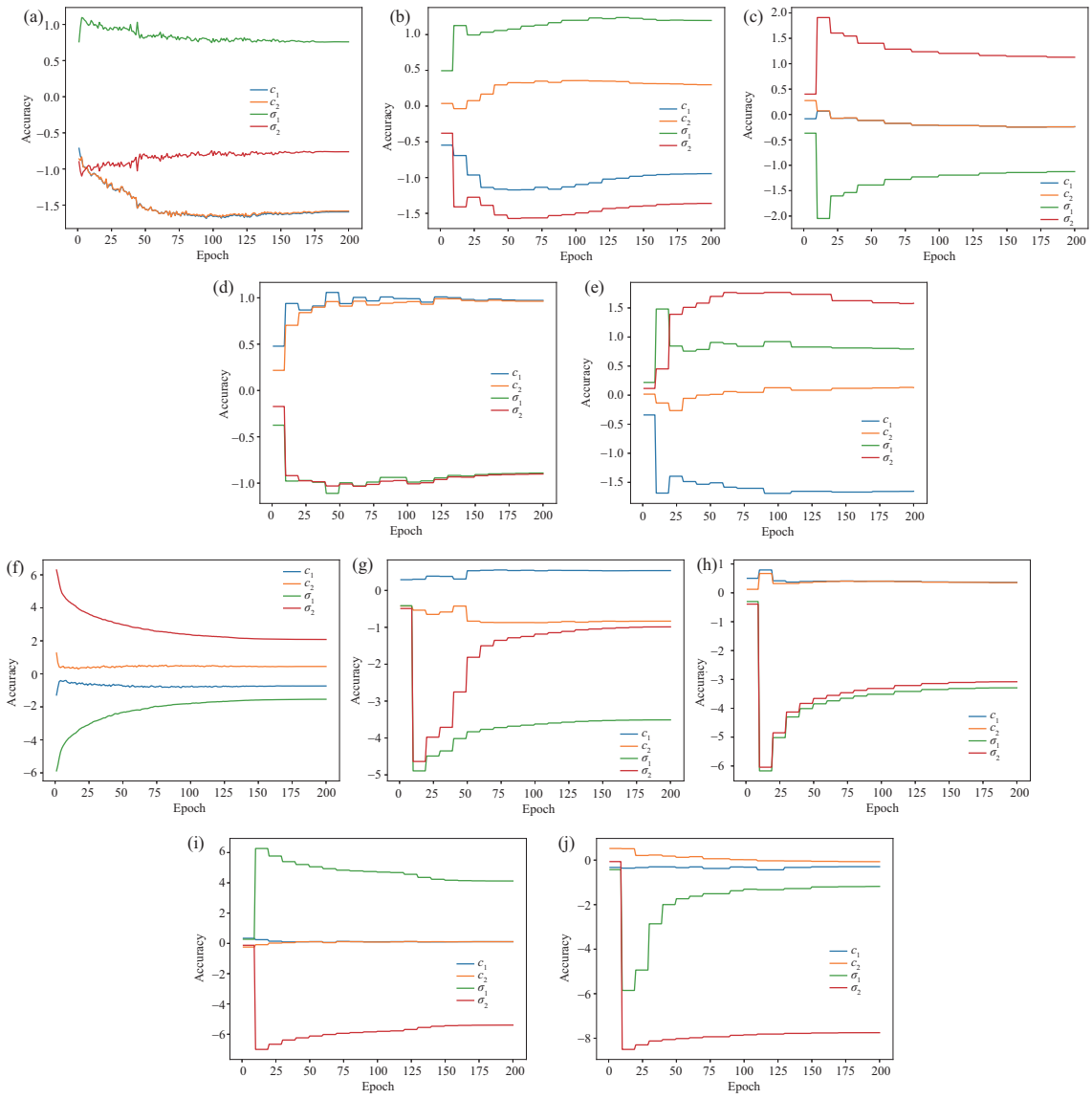

Figure 7 (Color online) Changing curves of parameters of the fuzzification layer with random initialization. (a) MIN-CFRNN; (b) MIN-CFRNN-FedAvg; (c) MIN-CFRNN-FedAdap; (d) MIN-CFRNN-FedAvg-UBDS; (e) MIN-CFRNN-FedAdap-UBDS; (f) MAX-CFRNN; (g) MAX-CFRNN-FedAvg; (h) MAX-CFRNN-FedAdap; (i) MAX-CFRNN-FedAvg-UBDS; (j) MAX-CFRNN-FedAdap-UBDS.

Table 6 Final values of the parameters of the randomly initialized fuzzification layer

		CFRNN	CFRNN-FedAvg	CFRNN-FedAdap	CFRNN-FedAvg-UBDS	CFRNN-FedAdap-UBDS
Minimum complexity	c_1	−1.594 159 01	−0.944 925 84	−0.239 974 07	0.972 042 32	−1.653 081 30
	c_2	−1.581 701 28	0.298 629 34	−0.241 450 29	0.962 891 28	0.130 313 63
	σ_1	0.759 116 83	1.197 505 83	−1.124 830 96	−0.890 909 43	0.796 333 61
	σ_2	−0.761 674 05	−1.360 974 43	1.127 511 02	−0.901 531 70	1.579 634 55
Maximum complexity	c_1	−0.734 762 79	0.537 959 64	0.363 144 55	0.111 743 58	−0.294 756 86
	c_2	0.448 872 77	−0.832 994 10	0.355 719 06	0.113 623 72	−0.070 333 49
	σ_1	−1.539 618 73	−3.509 645 22	−3.291 678 43	4.121 220 59	−1.180 452 35
	σ_2	2.085 323 57	−0.988 003 67	−3.080 972 43	−5.397 027 97	−7.750 771 52

(2) In [27], the global average pooling is replaced by the fuzzy layer, but in this study, the final expand layer is changed to the fuzzification layer, significantly reducing the parameter amount.

(3) In the supernet of this study, the connection weights in the rough layer and the output layer are transformed to positive values, which is not considered in [27].

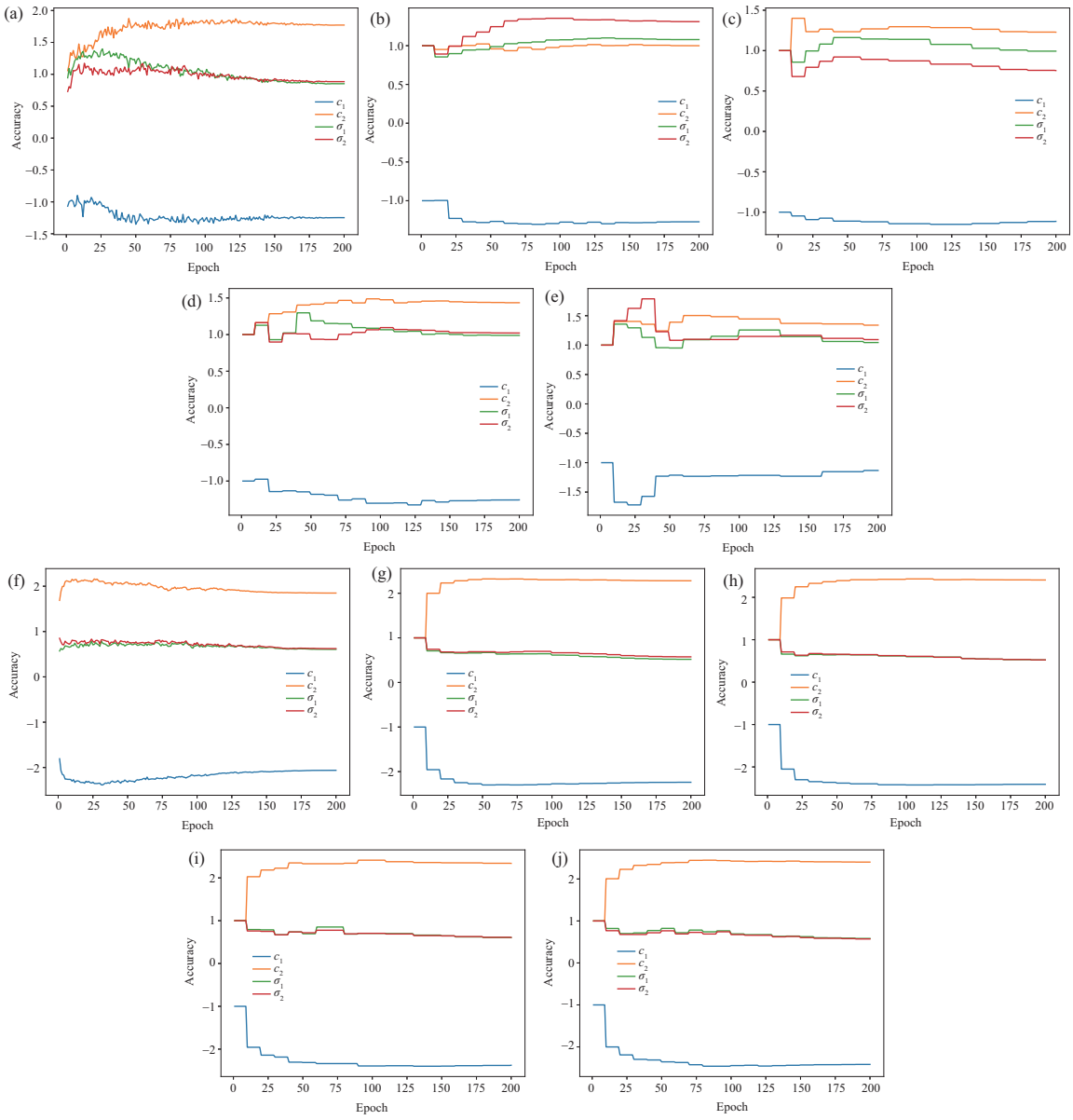

Figure 8 (Color online) Changing curves of parameters of the fuzzification layer with special initialization. (a) MIN-CFRNN; (b) MIN-CFRNN-FedAvg; (c) MIN-CFRNN-FedAdap; (d) MIN-CFRNN-FedAvg-UBDS; (e) MIN-CFRNN-FedAdap-UBDS; (f) MAX-CFRNN; (g) MAX-CFRNN-FedAvg; (h) MAX-CFRNN-FedAdap; (i) MAX-CFRNN-FedAvg-UBDS; (j) MAX-CFRNN-FedAdap-UBDS.

Though the proposed supernet architecture is lighter than that in [27], the cost of FL is approximately 0.06 of that in [27], and the test accuracies of "CFRNN-FedAdap" (99.86%) and "CFRNN-FedAdap-UBDS" (99.90%) are slightly worse than that of "max-FR-FedAvg" (99.96%) in [27].

Additionally, the proposed CFRNN in the FL environment is superior to other models aiming at LC25000, including the DHS-CapsNet [70], the multi-input capsule network [71], and a CNN model [72], the test accuracies of which are 99.23%, 99.58%, and 96.33%, respectively.

4.3.4 *Federated neuroevolution*

Similar to the supernet training, the neuroevolution is executed in the FedAdap environment. The parameter settings are $N_{\text{epoch}} = 5$, $N_{\text{epoch}}^{\text{th}} = 1$, $N_{\text{imp}}^{\text{th}} = 2$, and $N_{\text{stag}}^{\text{th}} = 2$. Figure 9 illustrates the Pareto front obtained by NSGANetV2 and the proposed memetic MOEA as well as the MOEA/D-1 in [27]. The horizontal and vertical coordinates denote the test error (%) and the complexity (FLOPs), respectively. In neuroevolution, the validation error is utilized as one objective. For illustration, we replace it with the

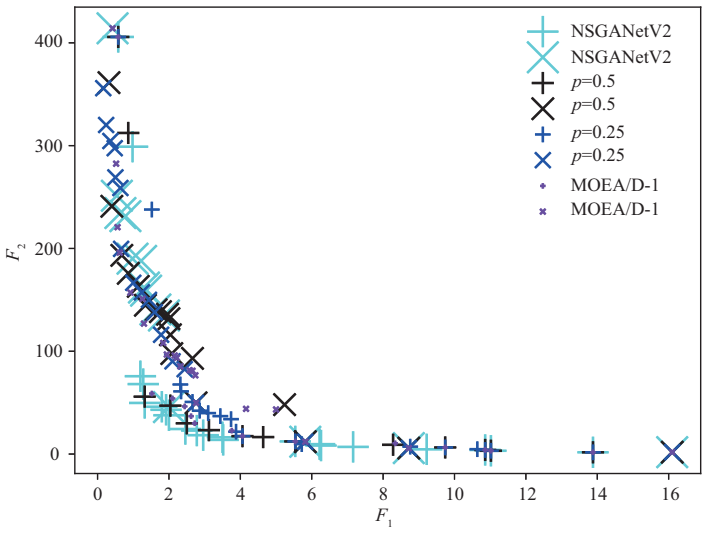

Figure 9 (Color online) Pareto front visualization.

test error. The aqua markers, black markers, blue markers, and blueviolet markers represent the Pareto fronts obtained using NSGANetV2, memetic MOEA with $p = 0.5$, memetic MOEA with $p = 0.25$, and MOEA/D-1, respectively. Besides the search space in Table 1, we also consider the use of CFRNN or CNN as a variable to optimize during neuroevolution. Therefore, we obtain two sets of Pareto fronts for each algorithm, which are denoted by + and × in Figure 9 for CFRNN and CNN, respectively. In general, CFRNN performs better for both objectives in the knee region (the lower left region), and CNN performs better for the objective of accuracy (the upper left region).

Among points in Pareto fronts, the solution in the upper left corner is a special solution of the first generation of solutions sampled during initialization, and its architecture is the same as that of the supernet. However, its classification error is not the same as in Table 3, owing to the fact that the weights of the supernet are saved in a file and loaded during neuroevolution. As the number of initial sampled subnets (100) is far greater than the number of subnets sampled during the next generations of evolution (8 subnets are trained for each generation), all MOEAs share the first-generation solutions. In addition, the weights of the supernet are inherited by the sampled subnets. It can be clearly seen that though the upper left black spot and blue spot have lower complexity than the upper left aqua spot and blueviolet spot, their classification errors are lower. In other words, by combining GA with NSGA-II, we can find network architectures with lower complexity and better classification performance than the supernet. In addition, the objective function values of each solution in all PFs are shown in Table 7. The optimal test accuracy obtained by DHS-CapsNet [70], the multi-input capsule network [71], and a CNN model [72] is 99.58%. In comparison, among the architectures optimized by NSGANetV2 and MOEA/D-1, only the initialized supernet has a higher test accuracy (99.60%) than 99.58%. Nevertheless, in the architectures obtained by the proposed memetic MOEA with $p = 0.5$, there are two architectures owning test accuracies above 99.58%, which are 99.68% and 99.60%. Among the architectures obtained by the proposed memetic MOEA with $p = 0.25$, the test accuracies of three architectures exceed 99.58% at 99.84%, 99.76%, and 99.64%.

To evaluate the performance of the proposed CFRNN architecture, another set of neuroevolutionary experiments has been performed. The experimental setup is the same as that of the previous set of neuroevolutionary experiments, the only difference is that all the architectures in this set of neuroevolutionary experiments are CFRNN. Pareto fronts obtained by experiments are shown in Figure 10, and the objective values of all solutions in Pareto fronts are listed in Table 8. For every solution generated by the benchmark algorithm NSGANetV2, a solution which is not dominated by it can be found in the solutions generated by the proposed algorithm. The proposed algorithm is also superior to the MOEA/D-1 algorithm. These results can be clearly observed from Table 8. Finally, in terms of the number of architectures with classification accuracy greater than 99.58%, NSGANetV2, the two proposed algorithms and MOEA/D-1 are 1, 3, 5, and 3, respectively, which is better than in the previous set of neuroevolutionary experiments.

Comparing the two sets of neuroevolutionary experiments, the proposed algorithm can generate more

Table 7 Objective function values of solutions in the Pareto front

NSGANetV2		$p = 0.5$		$p = 0.25$		MOEA/D-1	
Classification error (%)	FLOPs	Classification error (%)	FLOPs	Classification error (%)	FLOPs	Classification error (%)	FLOPs
0.42	414.13	0.32	361.29	0.16	355.89	0.42	414.13
0.54	251.28	0.4	241.02	0.24	320.09	0.52	282.45
0.64	234.46	0.68	193.29	0.36	304.5	0.56	220.67
0.78	231.41	0.86	175.8	0.48	297.52	0.62	195.94
0.98	190.05	–	–	0.5	269.09	0.92	157.14
–	–	–	–	0.64	258.8	–	–
–	–	–	–	0.66	199.58	–	–
1.2	75.7	1.14	163.14	1	166.59	1.26	150.99
1.28	68.05	1.32	55.85	1.24	157.28	1.3	126.92
1.32	49.72	–	–	1.44	149.98	1.52	58.87
1.8	46.02	–	–	1.6	137.93	–	–
1.92	43.07	–	–	1.78	116.02	–	–
2	37.72	2.04	47.1	2.1	90.19	2.1	54.09
2.46	24.35	2.5	29.89	2.32	67.67	2.44	46.32
2.78	22.83	–	–	2.34	60.96	2.62	36.82
2.96	18.33	–	–	2.66	50.7	2.72	29.93
–	–	–	–	2.76	49.82	–	–
–	–	–	–	2.86	42.16	–	–
3.5	16.29	3.12	23.26	3.1	39.82	3.76	22.83
3.52	13.81	4.06	17.4	3.44	36.88	4.06	17.4
5.54	12.29	4.64	16.51	3.74	33.94	5.54	12.29
5.82	11.97	5.54	12.29	3.88	21.74	5.82	11.97
6.22	9.61	5.82	11.97	4.06	17.4	–	–
6.26	8.27	–	–	5.54	12.29	–	–
7.16	6.97	–	–	5.72	9.64	–	–
8.72	5.78	8.28	9.04	8.72	5.78	8.34	10.44
9.22	4.59	8.72	5.78	10.64	4.72	8.72	5.78
10.86	3.99	10.86	3.99	10.86	3.99	10.86	3.99
11.02	3.22	11.02	3.22	11.02	3.22	11.02	3.22
13.88	1.69	13.88	1.69	13.88	1.69	13.88	1.69

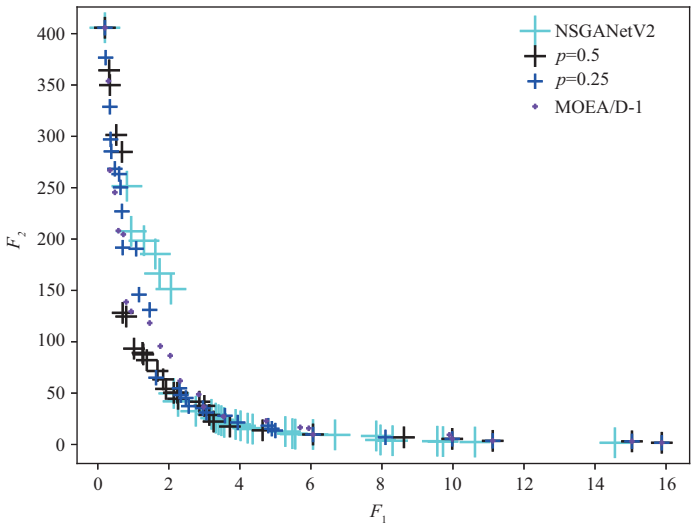

Figure 10 (Color online) Pareto front visualization of CFRNN.

solutions with a classification error less than 1% when the parameter $p = 0.25$. Although the classification error of the architecture with the highest accuracy obtained by the latter set of neuroevolutionary experiments is 0.04% higher than that of the former set, more architectures with a classification error

Table 8 Objective function values of solutions in the Pareto front of CFRNN

NSGANetV2		$p = 0.5$		$p = 0.25$		MOEA/D-1	
Classification error (%)	FLOPs	Classification error (%)	FLOPs	Classification error (%)	FLOPs	Classification error (%)	FLOPs
0.20	405.74	0.20	405.74	0.20	405.74	0.20	405.74
0.82	251.48	0.32	364.29	0.22	376.66	0.30	353.83
0.94	207.55	0.34	349.89	0.34	328.79	0.34	266.91
–	–	0.52	301.35	0.36	297.02	0.48	245.45
–	–	0.68	284.84	0.38	285.34	0.58	208.16
–	–	0.70	128.19	0.48	268.41	0.72	204.51
–	–	0.80	124.57	0.60	263.28	0.80	138.75
–	–	–	–	0.64	250.32	0.94	129.54
–	–	–	–	0.68	226.96	–	–
–	–	–	–	0.70	191.67	–	–
1.30	198.37	1.02	93.33	1.08	190.54	1.46	118.27
1.62	185.47	1.26	89.22	1.16	145.90	1.76	95.74
1.74	166.46	1.28	87.72	1.46	131.05	–	–
–	–	1.38	82.10	1.64	65.02	–	–
–	–	1.68	71.55	–	–	–	–
–	–	1.84	63.38	–	–	–	–
–	–	1.92	54.13	–	–	–	–
2.06	151.31	2.14	50.37	2.30	54.71	2.04	86.48
2.14	49.72	2.24	50.36	2.34	48.07	2.32	61.97
2.26	41.99	2.26	44.52	2.48	45.29	2.84	48.78
2.76	32.39	2.86	41.64	2.56	37.32	2.98	36.64
–	–	–	–	2.96	35.34	–	–
3.20	29.39	3.00	37.58	3.06	31.11	3.54	27.16
3.32	25.65	3.14	28.73	3.58	27.98	4.74	22.94
3.40	24.07	3.26	22.32	3.94	21.41	5.70	16.61
3.48	23.12	3.72	17.49	4.80	18.35	5.94	15.74
3.56	21.56	4.64	13.78	4.90	15.38	6.06	9.73
3.88	19.52	6.06	9.73	5.00	13.58	–	–
4.02	18.15	–	–	6.06	9.73	–	–
4.22	15.90	–	–	–	–	–	–
4.36	15.02	–	–	–	–	–	–
5.28	12.57	–	–	–	–	–	–
5.48	10.52	–	–	–	–	–	–
5.56	10.08	–	–	–	–	–	–
6.06	9.73	–	–	–	–	–	–
6.68	9.34	–	–	–	–	–	–
7.84	8.29	–	–	–	–	–	–
7.96	4.40	–	–	–	–	–	–
8.30	3.67	8.62	6.93	8.10	6.94	9.90	9.41
9.56	3.19	9.98	5.46	9.98	5.46	9.98	5.46
9.72	2.78	11.12	3.55	11.12	3.55	11.12	3.55
10.62	2.38	15.04	3.04	15.04	3.04	15.04	3.04
14.56	1.68	15.88	1.69	15.88	1.69	15.88	1.69

of less than 1% have been obtained. In addition, in the latter set of neuroevolutionary experiments, the proposed algorithm performs better in the lower left knee area.

4.3.5 *Extension to a more complex dataset*

In the LC25000 dataset [68], the original images include 750 images of lung tissue and 500 images of colon tissue. There are five categories in total. Based on these 1250 images, 25000 images were incrementally generated, with each category increasing from 250 images to 5000 images. That is 20 copies of each image. The probability that any two images selected from the dataset are from the same original image is $6.08\times10^{-5}\%$. Therefore, in the data partitioning of FL in the previous experimentation, there is a certain

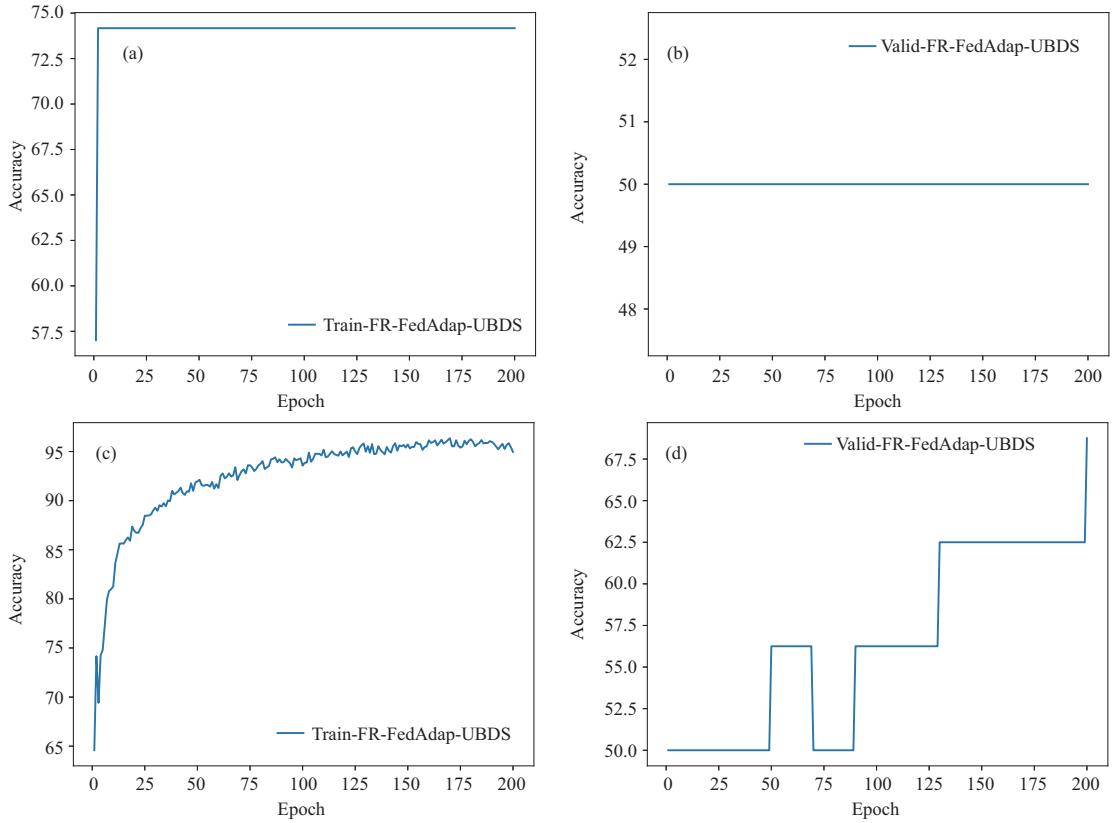

Figure 11 (Color online) Training and validation accuracy curves of the Chest X-ray images (Pneumonia) dataset. (a) MIN training; (b) MIN validation; (c) MAX training; (d) MAX validation.

Table 9 Best training and validation accuracies of the minimum- and maximum-complexity models and corresponding epoch numbers as well as final test accuracies with respect to the Chest X-ray images (Pneumonia) dataset

	Best acc	Epoch #	Test acc
Minimum complexity	74.17 (50.00)	46 (0)	62.50 (200)
Maximum complexity	96.35 (68.75)	169 (199)	92.15 (200)

correlation between different data partitions. For this purpose, Chest X-Ray Images (Pneumonia) [73], another medical image dataset, is utilized for experimentation. There are two categories of images in the dataset, which are normal and pneumonia, and the images are divided into fixed training, validation, and test sets, in which, the number of images for the two categories is 1341 and 3875, 8 and 8, as well as 234 and 390, respectively. There is almost no correlation between the different images in the dataset. Compared to the LC25000 dataset, this dataset is better able to test the performance of the proposed FL framework. For the experimentation based on the Chest X-ray images (Pneumonia) dataset, the parameter settings are the same as those for the previous experimentation. The accuracy curves are illustrated in Figure 11 and the accuracy values are listed in Table 9.

In this part of the experiment, one network architecture, namely CFRNN, is considered, and the architectures with the minimum complexity and the maximum complexities are trained. For the data partitioning, only the UBDS partitioning is considered, and the partitioning ratio was the same as in previous experiments. In FL strategy, only FedAdap scheme is used. For the minimum-complexity architecture, on the training dataset (Figure 11(a)), the accuracy of CFRNN has been consistently around 74%. On the validation dataset (Figure 11(b)), CFRNN consistently achieves 50% accuracy. The validation accuracy is not high, which may be due to the small number of images in the validation dataset (only 8 images per category). For the supernet, on the training dataset (Figure 11(c)), the performance of CFRNN is increasing throughout the training process, and in the end, CFRNN achieves nearly 95% accuracy. On the validation dataset (Figure 11(d)), the performance of CFRNN fluctuates greatly. Finally, CFRNN reaches an accuracy of approximately 70%.

As can be seen from Table 9, although the verification accuracy is lower than the training accuracy,

the test accuracy is not very low for the architecture with the least complexity. For supernets, validation accuracy is still significantly lower than training accuracy, but test accuracy is very close to training accuracy. The network architecture in [73] achieved an accuracy of 92.8%. Although the accuracy of CFRNN supernet in this study is 92.15%, as the Inception V3 architecture was adopted in [73], its complexity is far higher than that of the MobileNetV3 architecture adopted in this study. Therefore, in summary, the CFRNN architecture can simultaneously take into account the accuracy based on the architecture with lower complexity, in the FedAdap environment.

5 Conclusion

The 5G-and-beyond communication systems have facilitated the improvement of network automation and intelligent systems. Based on the next-generation communication network, an adaptive interpretable FL framework with neuroevolution for mobile telemedicine systems is proposed in this paper. As there exist multiple parameters in such a DL model, an adaptive FL, denoted as FedAdap, is developed to reduce the communication burden in FL and thus update the DL model based on the training status. To strengthen the interpretability of DL, a CFRNN model is proposed through incorporating fuzzy rough set theory in place of the final expand layer, the global average pooling, the feature mix layer, and the classifier layer, acting as the further processor after the feature learning via traditional convolutional layer. Based on the trained CFRNN supernet, the neuroevolution is performed to explore light-weight networks with high accuracy. By combining a local search of the high-accuracy architecture with an MOEA simultaneously to tackle accuracy and complexity, a memetic MOEA is developed to achieve higher performance. The supernet training and neuroevolution are both executed in the FedAdap environment, guaranteeing the security and privacy of medical imaging. To test the data heterogeneity, an unbalanced data distribution is considered. It is seen that the adaptive interpretable FL framework with neuroevolution for medical imaging is conducive to the research and application relating to mobile intelligent systems.

Acknowledgements This work was supported by Natural Science Fund of Hebei Province for Distinguished Young Scholars (Grant No. F2021202010), Science and Technology Project of Hebei Education Department (Grant No. JZX2023007), and S&T Program of Hebei (Grant No. 225676163GH).

References

1 Xu J D, Yuen C, Huang C W, et al. Reconfiguring wireless environments via intelligent surfaces for 6G: reflection, modulation, and security. Sci China Inf Sci, 2023, 66: 130304
2 Sheng M, Zhou D, Bai W G, et al. Coverage enhancement for 6G satellite-terrestrial integrated networks: performance metrics, constellation configuration and resource allocation. Sci China Inf Sci, 2023, 66: 130303
3 Sheller M J, Reina G A, Edwards B, et al. Multi-institutional deep learning modeling without sharing patient data: a feasibility study on brain tumor segmentation. In: Proceedings of Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries, 2019. 92–104
4 Lee H, Chai Y J, Joo H, et al. Federated learning for thyroid ultrasound image analysis to protect personal information: validation study in a real health care environment. JMIR Med Inform, 2021, 9: e25869
5 Fan Z, Su J, Gao K, et al. A federated deep learning framework for 3D brain MRI images. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN), 2021
6 Feki I, Ammar S, Kessentini Y, et al. Federated learning for COVID-19 screening from Chest X-ray images. Appl Soft Computing, 2021, 106: 107330
7 Baheti P, Sikka M, Arya K, et al. Federated learning on distributed medical records for detection of lung nodules. In: Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, 2020
8 Zhang W S, Zhou T, Lu Q H, et al. Dynamic-fusion-based federated learning for COVID-19 detection. IEEE Internet Things J, 2021, 8: 15884–15891
9 Sakib S, Fouda M M, Fadlullah Z M, et al. On COVID-19 prediction using asynchronous federated learning-based agile radiograph screening booths. In: Proceedings of the IEEE International Conference on Communications (ICC 2021), 2021
10 Zhang Z, Zhang Y, Guo D, et al. Communication-efficient federated continual learning for distributed learning system with Non-IID data. Sci China Inf Sci, 2023, 66: 122102
11 Xu H, Li J, Xiong H, et al. FedMax: enabling a highly-efficient federated learning framework. In: Proceedings of the IEEE 13th International Conference on Cloud Computing (CLOUD), 2020
12 Shin H C, Roth H R, Gao M, et al. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. IEEE Trans Med Imag, 2016, 35: 1285–1298
13 Li Y, Liu D, Li H Q, et al. Learning a convolutional neural network for image compact-resolution. IEEE Trans Image Process, 2019, 28: 1092–1107
14 Arrieta A B, Díaz-Rodríguez N, Ser J D, et al. Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. Inf Fusion, 2020, 58: 82–115
15 Zhang Q S, Wu Y N, Zhu S C. Interpretable convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018
16 Saeed T, Loo C K, Kassim M S S. Ensembles of deep learning framework for stomach abnormalities classification. Comput Mater Continua, 2022, 70: 4357–4372

17 Zeiser F A, da Costa C A, Ramos G O, et al. DeepBatch: a hybrid deep learning model for interpretable diagnosis of breast cancer in whole-slide images. Expert Syst Appl, 2021, 185: 115586
18 Lu J. Research on Rough Modeling of Type-2 Fuzzy Sets. Dissertation for Ph.D. Degree. Taiyuan: Shanxi University, 2018
19 Ji W T, Pang Y, Jia X Y, et al. Fuzzy rough sets and fuzzy rough neural networks for feature selection: a review. WIREs Data Min Knowl, 2021, 11: e1402
20 Hussain K, Salleh M N M, Cheng S, et al. Metaheuristic research: a comprehensive survey. Artif Intell Rev, 2019, 52: 2191–2233
21 Ser J D, Osaba E, Molina D, et al. Bio-inspired computation: where we stand and what's next. Swarm Evolary Computation, 2019, 48: 220–250
22 Cao B, Zhao J W, Lv Z H, et al. Multiobjective evolution of fuzzy rough neural network via distributed parallelism for stock prediction. IEEE Trans Fuzzy Syst, 2020, 28: 939–952
23 Cao B, Zhao J W, Liu X, et al. Multiobjective evolution of the explainable fuzzy rough neural network with gene expression programming. IEEE Trans Fuzzy Syst, 2022, 30: 4190–4200
24 Yeganejou M, Dick S, Miller J. Interpretable deep convolutional fuzzy classifier. IEEE Trans Fuzzy Syst, 2020, 28: 1407–1419
25 Operiano K R G, Iba H, Pora W. Neuroevolution architecture backbone for X-ray object detection. In: Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI), 2020
26 Zhu H Y, Zhang H Y, Jin Y C. From federated learning to federated neural architecture search: a survey. Complex Intell Syst, 2021, 7: 639–657
27 Liu X, Zhao J W, Li J, et al. Federated neural architecture search for medical data security. IEEE Trans Ind Inf, 2022, 18: 5628–5636
28 Lu Z C, Deb K, Goodman E, et al. NSGANetV2: evolutionary multi-objective surrogate-assisted neural architecture search. In: Proceedings of the 16th European Conference on Computer Vision, 2020
29 Li H Y, Chen D D, Nailon W H, et al. Dual convolutional neural networks for breast mass segmentation and diagnosis in mammography. IEEE Trans Med Imag, 2022, 41: 3–13
30 Ahmadian S, Jalali S M J, Islam S M S, et al. A novel deep neuroevolution-based image classification method to diagnose coronavirus disease (COVID-19). Comput Biol Med, 2021, 139: 104994
31 Hassanzadeh T, Essam D, Sarker R. Evolutionary deep attention convolutional neural networks for 2D and 3D medical image segmentation. J Digit Imag, 2021, 34: 1387–1404
32 Brendan McMahan H, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, 2017
33 Nguyen D C, Ding M, Pathirana P N, et al. Federated learning for Internet of Things: a comprehensive survey. IEEE Commun Surv Tutorials, 2021, 23: 1622–1658
34 Chen H, Li H, Xu G, et al. Achieving privacy-preserving federated learning with irrelevant updates over e-health applications. In: Proceedings of IEEE International Conference on Communications (ICC), 2020
35 Xu X H, Peng H, Bhuiyan M Z A, et al. Privacy-preserving federated depression detection from multisource mobile health data. IEEE Trans Ind Inf, 2022, 18: 4788–4797
36 Ge K S, Zhang Y M, Fu Y Q, et al. Accelerate distributed deep learning with cluster-aware sketch quantization. Sci China Inf Sci, 2023, 66: 162102
37 Zhou X K, Yang Q Y, Liu Q, et al. Spatial-temporal federated transfer learning with multi-sensor data fusion for cooperative positioning. Inf Fusion, 2024, 105: 102182
38 Cheng Y Y, Lu J Y, Niyato D, et al. Federated transfer learning with client selection for intrusion detection in mobile edge computing. IEEE Commun Lett, 2022, 26: 552–556
39 Zhu H Y, Jin Y C. Multi-objective evolutionary federated learning. IEEE Trans Neural Netw Learn Syst, 2020, 31: 1310–1322
40 Zhu G X, Lyu Z H, Jiao X, et al. Pushing AI to wireless network edge: an overview on integrated sensing, communication, and computation towards 6G. Sci China Inf Sci, 2023, 66: 130301
41 Tam P, Math S, Nam C, et al. Adaptive resource optimized edge federated learning in real-time image sensing classifications. IEEE J Sel Top Appl Earth Observations Remote Sens, 2021, 14: 10929–10940
42 Zong R X, Qin Y C, Wu F, et al. Fedcs: efficient communication scheduling in decentralized federated learning. Inf Fusion, 2024, 102: 102028
43 Wu L M, Jin Y C, Hao K R. Optimized compressed sensing for communication efficient federated learning. Knowledge-Based Syst, 2023, 278: 110805
44 Chen S H, Jie Z Y, Wang G J, et al. A new federated learning-based wireless communication and client scheduling solution for combating COVID-19. Comput Commun, 2023, 206: 101–109
45 Zhou C S, Ansari N. Securing federated learning enabled NWDAF architecture with partial homomorphic encryption. IEEE Netw Lett, 2023, 5: 299–303
46 Li J, Wei K, Ma C, et al. DP-GenFL: a local differentially private federated learning system through generative data. Sci China Inf Sci, 2023, 66: 189303
47 Kong L X, Zheng G, Brintrup A. A federated machine learning approach for order-level risk prediction in supply chain financing. Int J Production Economics, 2024, 268: 109095
48 Truhn D, Arasteh S T, Saldanha O L, et al. Encrypted federated learning for secure decentralized collaboration in cancer image analysis. Med Image Anal, 2024, 92: 103059
49 Wu G J, Li J, Ning Z L, et al. Federated learning enabled credit priority task processing for transportation big data. IEEE Trans Intell Transp Syst, 2024, 25: 839–849
50 Hu X C, Qin J D, Shen Y H, et al. An efficient federated multiview fuzzy C-means clustering method. IEEE Trans Fuzzy Syst, 2024, 32: 1886–1899
51 Zhang L J, Shi Y, Chang Y-C, et al. Federated fuzzy neural network with evolutionary rule learning. IEEE Trans Fuzzy Syst, 2023, 31: 1653–1664
52 Vinita L J, Vetriselvi V. Federated learning-based misbehaviour detection on an emergency message dissemination scenario for the 6G-enabled Internet of Vehicles. Ad Hoc Networks, 2023, 144: 103153
53 Yoo E, Ko H, Pack S. Fuzzy clustered federated learning algorithm for solar power generation forecasting. IEEE Trans Emerg Top Comput, 2022, 10: 2092–2098
54 Przybyła-Kasperek M, Opoku K. Decision rules for dispersed data using a federated learning approach. Procedia Comput Sci, 2023, 225: 4305–4313
55 Hu Y, Zhang Y, Gong D W, et al. Multiparticipant federated feature selection algorithm with particle swarm optimization

for imbalanced data under privacy protection. IEEE Trans Artif Intell, 2023, 4: 1002–1016
56 Pedrycz W. Design, interpretability, and explainability of models in the framework of granular computing and federated learning. In: Proceedings of IEEE Conference on Norbert Wiener in the 21st Century, 2021. 1–6
57 Sarkar M, Yegnanarayana B. Fuzzy-rough membership functions. In: Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, 1998
58 Ye Y. Structure and parameters optimization of fuzzy rough neural network. Syst Eng Electron, 2009, 31: 2988–2993
59 Xu S Y, Qin K Y, Pan X D, et al. Rough set model based on axiomatic fuzzy set. J Intell Fuzzy Syst, 2023, 45: 1423–1436
60 Qi G A, Yang B, Li W. Some neighborhood-related fuzzy covering-based rough set models and their applications for decision making. Inf Sci, 2023, 621: 799–843
61 Zhao J W, Cao B, Liu X, et al. Multiobjective multiple mobile sink scheduling via evolutionary fuzzy rough neural network for wireless sensor networks. IEEE Trans Fuzzy Syst, 2022, 30: 4630–4641
62 Liu X, Zhao J W, Li J, et al. Large-scale multiobjective federated neuroevolution for privacy and security in the Internet of Things. IEEE Internet Things M, 2022, 5: 74–77
63 Liu X, Li J, Zhao J W, et al. Evolutionary neural architecture search and its applications in healthcare. Comput Modeling Eng Sci, 2024, 139: 143–185
64 Stanley K O, Clune J, Lehman J, et al. Designing neural networks through neuroevolution. Nat Mach Intell, 2019, 1: 24–35
65 Khan S, Rizwan A, Khan A N, et al. A multi-perspective revisit to the optimization methods of neural architecture search and hyper-parameter optimization for non-federated and federated learning environments. Comput Electrical Eng, 2023, 110: 108867
66 Zhu H Y, Jin Y C. Real-time federated evolutionary neural architecture search. IEEE Trans Evol Computat, 2022, 26: 364–378
67 Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Computat, 2002, 6: 182–197
68 Borkowski A A, Bui M M, Thomas L B, et al. Lung and colon cancer histopathological image dataset (LC25000). 2019. arXiv:1912.12142v1
69 Cao T D, Truong-Huu T, Tran H, et al. A federated deep learning framework for privacy preservation and communication efficiency. J Syst Architecture, 2022, 124: 102413
70 Adu K, Yu Y B, Cai J Y, et al. DHS-CapsNet: dual horizontal squash capsule networks for lung and colon cancer classification from whole slide histopathological images. Intl J Imaging Syst Technol, 2021, 31: 2075–2092
71 Ali M, Ali R. Multi-input dual-stream capsule network for improved lung and colon cancer classification. Diagnostics, 2021, 11: 1485
72 Masud M, Sikder N, Nahid A A, et al. A machine learning approach to diagnosing lung and colon cancer using a deep learning-based classification framework. Sensors, 2021, 21: 748
73 Kermany D S, Goldbaum M, Cai W J, et al. Identifying medical diagnoses and treatable diseases by image-based deep learning. Cell, 2018, 172: 1122–1131