Client Selection Method for Federated Learning in Multi-robot Collaborative Systems

Nian Ding[1,3](✉), Chunrong Peng[2], Min Lin[3], Yangfei Lin[1], Zhaoyang Du[1], and Celimuge Wu[1]

[1] Graduate School of Informatics and Engineering, The University of Electro-Communications, 1-5-1 Chofugaoka, Chofu, Tokyo 182-8585, Japan
3192720267@qq.com, celimuge@uec.ac.jp

[2] Library, Inner Mongolia University of Finance and Economics, Xincheng District, Hohhot, Inner Mongolia Autonomous Region, People's Republic of China

[3] College of Computer Science and Technology, Inner Mongolia Normal University, Saihan District, Hohhot, Inner Mongolia Autonomous Region, People's Republic of China

Abstract. Federated Learning (FL) has recently attracted considerable attention in multi-robot collaborative systems, owning to its capability of enabling mobile clients to collaboratively learn a global prediction model without sharing their privacy-sensitive data to the server. In a multi-robot collaboration system, an approach that ensures privacy-preserving knowledge sharing among multiple robots becomes imperative. However, the application of FL in such systems encounters two major challenges. Firstly, it is inefficient to use all the network nodes as federated learning clients (which conduct training of machine learning model based on own data) due to the limited wireless bandwidth and energy of robots. Secondly, the selection of an appropriate number of clients must be carefully considered, considering the constraints imposed by limited communication resources. Selecting an excessive number of clients may result in a failure in uploading important models. To overcome these challenges, this paper proposes a client selection approach that considers multiple metrics including the data volume, computational capability, and network environment by integrating fuzzy logic and Q-learning. The experimental results validate the theoretical feasibility of the proposed approach. Further empirical data can be derived from training experiments on public datasets, enhancing the practical applicability of the proposed method.

Keywords: Federated Learning · Fuzzy Logic · Q-learning Algorithm · Multi-robots Collaboration

1 Introduction

Artificial intelligence (AI) has become an indispensable element of modern society, owing to the recent advancements in machine learning (ML) technology that have enabled successful implementations in various domains, such as image recognition and natural language processing [1]. ML models require a large amount of data to update the

C. Wu et al. (Eds.): MONAMI 2023, LNICST 559, pp. 27–41, 2024.
https://doi.org/10.1007/978-3-031-55471-1_3

model to achieve the desirable accuracy [2]. In traditional ML model training, a cloud centric approach is adopted to gather data collected by mobile devices and train the ML model on a powerful cloud server or data center [3]. However, the data transmission process raises security concerns, as it increases the risk of data breaches and compromises user privacy when data is aggregated in the cloud. The current trend in machine learning is shifting towards decentralized edge clouds, leading to the emergence of FL. FL allows the efficient training and utilization of ML models while ensuring user data localization [4, 5].

In order to protect the security of local data for each robot, prevent the leakage of important or sensitive data, and achieve information sharing between robots, FL has been applied in the field of multi-robot collaboration. Unlike the independent and identically distributed (IID) data environment of traditional machine learning, FL operates in a complex and heterogeneous environment, with significant differences in sample size and data distribution among different clients. This indicates that there is a Non-Independent and Identical Distribution (Non-IID) feature between the local data of the client in FL [6]. There are two key challenges in the application of FL in multi-robots collaborative systems. Firstly, the local data of each robot (client) is inconsistent in practical applications, and some robots may not be helpful or even ineffective for collaborative systems. Hence, it is crucial to select clients based on their performance to optimize information exchange. Selecting robots with better performance for information sharing is currently a challenging and active research area. Secondly, in multi-robots collaborative systems, communication between the clients (robots) and the central server occurs via network connections, which may be unstable and prone to temporary disconnections. Due to the limitations of network resources, the selection of an appropriate number of clients for information exchange becomes critical. Choosing an optimal client count depends on factors such as network status, computational resource capabilities, and FL model accuracy. Consequently, determining the appropriate number of clients for information exchange represents another research challenge and focal point. To address the above two challenges, this paper proposes a client selection method based on fuzzy logic and Q-learning for FL in multi-robot collaborative systems. The purpose of the paper is to select high-performance clients based on the local data volume and computing resource capabilities and to select an appropriate number of clients based on network status, computing resource capabilities and model accuracy of FL. The main contributions of the paper are listed below:

- The paper proposes a client evaluation approach based on fuzzy logic to tackle the challenge of selecting clients with better performance. This approach considers the data volume and computational capability of each client. By setting the corresponding fuzzy combination based on local data size and model training time, the approach obtains fuzzy results through fuzzification and fuzzy reasoning. These fuzzy results are then converted into numerical values, enabling the evaluation of all candidate clients. By utilizing a fuzzy logic algorithm, the proposed approach effectively selects clients with superior performance.
- The paper addresses the problem of determining the appropriate number of clients for FL by proposing an approach based on the Q-learning algorithm. The Q-table and R matrix are initialized, and the initial action is chosen based on the R matrix

and initial state. Subsequently, the Q-table is iteratively updated using feedback from the environment until the appropriate number of clients is determined. This approach contributes to client selection by finding the optimal number of clients, taking into account factors such as network status, computational resource capabilities, and FL model accuracy.

- To verify the effectiveness of the proposed scheme, this paper conducts empirical training experiments on GTSRB (The German Traffic Sign Benchmark). The simulation results show that the proposed scheme outperforms other baseline methods.

The remainder of the paper is organized as follows. Related works are listed in Sect. 2. The overview of the system model is introduced in Sect. 3. A method for selecting federated learning clients based on fuzzy logic is described in Sect. 4. A method for selecting federated learning clients based on Q-learning is described in Sect. 5. Following, we present the performance of the proposed scheme using a simulator in Sect. 6. Finally, we conclude our work in Sect. 7.

2 Related Work

FL has gained significant attention and widespread application in multi-robot collaborative systems, as it enables collaborative machine learning without the need for centralized data storage [7]. FL can use data collected by a set of robots distributed in different locations to train ML models. Reference [8] studies a new federated deep reinforcement learning method called F-DRL. This method addresses issues such as signal strength attenuation caused by obstacles and the dynamic environment caused by robot movement. By utilizing a dynamic long-term goal, each robot autonomously plans its own path and downlink power, reducing training dimensions and saving computational costs [8]. Reference [9] proposes a distributed federated learning method for networked multi-robots called dFRL. This method solves the inherent problems of centralized federated learning methods such as central node failures and channel bandwidth bottlenecks. Unlike traditional centralized methods that rely on a limited number of cloud servers to aggregate models, dFRL performs model aggregation through parameter transfer among robots, providing a decentralized and scalable approach [9].

Another key issue is that a group of robots typically collect data at different rates, which affects the frequency of robot participation in shared model updates. Reference [10] proposes a data-driven approach where robots update their local model only when sufficient data has been collected. However, an important challenge in FL for multi-robots systems lies in selecting the most suitable client to participate in the training process [10]. Because each client may have different data distribution and computational capability, this may affect the overall performance of the model. McMahan et al. reveal that the convergence performance of FL depends on the importance of client updates [11]. In the context of client selection, reference [12] investigates the problem from the perspective of minimizing average model exchange time. Meanwhile, reference [13] points out that selecting clients with higher local loss can achieve faster convergence in the learning process. They propose an efficient client selection method that considers factors such as convergence speed, result bias and communication/computation cost [13]. Furthermore, FL is applied to multi-robots reinforcement learning in reference [14]. They focus on establishing a collaborative model in the context of 5G HetNet, which can quickly and stably complete tasks while ensuring security performance [14].

Although these studies have contributed to addressing various challenges in FL for multi-robot systems, there are still opportunities for improvement in terms of efficiency, scalability, and robustness. This paper focuses on investigating client selection methods for federated learning in multi-robots collaborative systems, aiming to enhance the quality of trained models and reduce communication overhead.

3 System Overview

3.1 Task Scenario

As shown in Fig. 1, multiple robots collaborate to move goods in a multi-robots collaborative system. Robots collaborate through navigation technology to transport goods from point A to point B. To enhance their navigation capabilities, traffic signs are placed on both sides of the road.

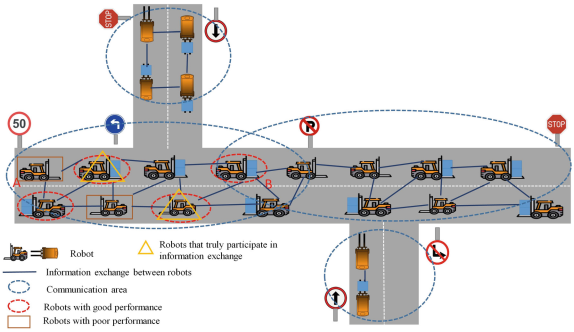

Fig. 1. Application scenario of FL client selection method for multi-robots collaborative systems.

In order to preserve the security of local data, FL is used for information exchange in a multi-robots collaborative system. Choosing some useless or weakly correlated robots as clients for FL may waste communication resources and lead to a decrease in the performance of the global model. It is necessary to select clients with better performance to participate in FL, which can help improve performance of FL. Due to network resource constraints, different clients' numbers are selected based on the requirements of different scenarios, such as selecting 20, 40, 60 or 80 clients for information exchange. It is necessary to select high-performance robots for information exchange while also choosing an appropriate robots' numbers according to different scenarios requirements.

3.2 The Workflow of Fuzzy Logic

We have designed a FL client method based on fuzzy logic. As shown in Fig. 2, it mainly consists of the following two modules:

- Clients' local training. Each client trains a local model using its local dataset. The client's local data volume (LDV) and resource computation time (local model training time, RCT) serve as two factors that are considered together to assess the client's performance.
- Fuzzy logic model. The client's LDV and RCT are used as inputs. Then, fuzzy results are obtained through fuzzy inference. The final client selection result is obtained through defuzzification.

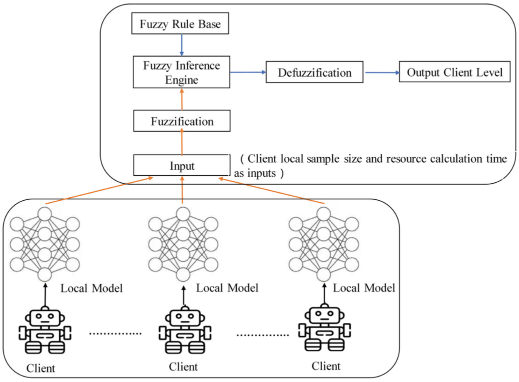

Fig. 2. Workflow of fuzzy logic.

3.3 The Workflow of Q-Learning

The proposed FL client selection method incorporates the utilization of Q-learning.

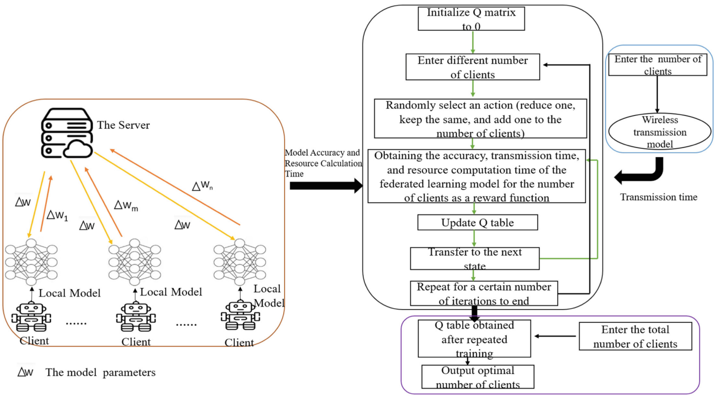

Fig. 3. Workflow of Q-learning.

As shown in Fig. 3, it consists of the following three key modules:

- FL model. FL accuracy (ACC) and clients' RCT are obtained through FL are used as inputs for the Q-learning algorithm.
- Wireless transmission model (WT). In FL, model parameters are transmitted between the client and server through WT. The obtained transmission time through WT is used as an input for the Q-learning algorithm.
- Q-learning algorithm. Obtain the final Q-table. Enter a total clients' numbers. Output an optimal clients' numbers.

4 A Method for Selecting Federated Learning Clients Based on Fuzzy Logic

4.1 Fuzzy Set and Membership Functions

The comprehensive performance of a client is evaluated based on two aspects: the client's LDV and its RCT. According to the comprehensive performance of the client, it is divided into three levels: high, middle, and low. There are two fuzzy sets, namely LDV set 'Num' and RCT set 'Time'. The set 'Num' is {many, middle, few}, and the set 'Time' is {high, middle, low}. The client data is Non-IID, with a maximum of 1200 images and a minimum of 100 images for LDV. The interval [0, 500] is defined as the element "few" in set 'Num'. [300, 900] is defined as the element 'middle' in set 'Num'. [700, 1200] is defined as the element 'many' in set 'Num'. [64, 68] is defined as the element 'low' in set 'Time'. [66, 62] is defined as the element 'middle' in set 'Time'. [70, 74] is defined as the element 'high' in set 'Time'.

When the value of LDV belongs to the set 'Num', the membership function is defined as follows:

$$F_{Num}(X)\begin{cases} 1/(1+e^{-a(x-c)}) & x \in [0, 500] \\ 1/(1+e^{-a((x-300)-c)}) & x \in [300, 900] \\ 1/(1+e^{-a((x-700)-c)}) & x \in [700, 1200] \end{cases} \tag{1}$$

a and c are feature parameters. x represents the values of the set 'Num' within the range [0, 1200]. As shown in Fig. 4, the membership function of set 'Num' for values within the range [0, 1200].

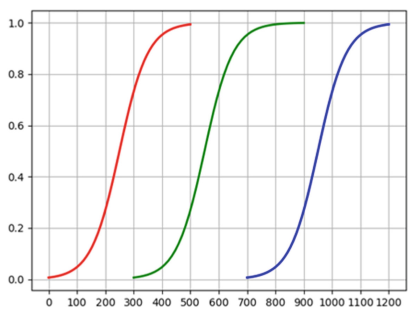

Fig. 4. The membership function of the set 'Num' for values within the range [0, 1200].

When the value of the element ‘low’ in the set ‘Time’ is [64, 68], the membership function is defined as follows:

$$F_{Time}(X) = (\frac{x-s}{b-s})^4 \tag{2}$$

s and b are feature parameters. x represents the values of the set ‘Time’ within the range [64, 68].

When the value of the element ‘middle’ belongs to the set ‘Time’, the membership function is defined as follows:

$$F_{Time}(X) \begin{cases} (\frac{x-c}{d-c})^4 & x \in [66, 68] \\ \frac{x}{x} & x \in [68, 70] \\ (\frac{t-x}{t-u})^4 & x \in [70, 72] \end{cases} \tag{3}$$

d, c, t, and u are feature parameters. x represents the values of the set ‘Time’ within the range [66, 72].

When the value of the element ‘high’ belongs to the set ‘Time’, the membership function is defined as follows:

$$F_{Time}(X) \begin{cases} \frac{x}{x} & x \in [70, 72] \\ (\frac{p-x}{p-k})^4 & x \in [72, 74] \end{cases} \tag{4}$$

p and k are feature parameters. x represents the values of the set ‘Time’ within the range [70, 74].

As shown in Fig. 5, the membership function of the set ‘Time’ for values within the range [64, 74].

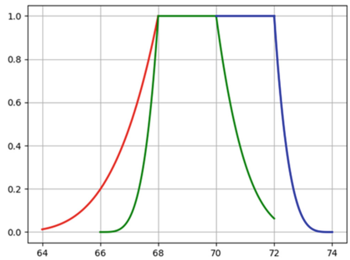

Fig. 5. The membership function of the set ‘Time’ for values within the range [64, 74].

4.2 Fuzzification and Defuzzification

For the set ‘Num’ {few, middle, many}, each element in ‘Num’ corresponds to a membership degree. Define them as N_0, N_1 and N_2 in sequence. For the set ‘Time’ {low, middle, high}, each element in ‘Time’ corresponds to a membership degree. Define them as T_0, T_1 and T_2 in sequence. The combination of the value from ‘Num’ and the value

Table 1. Fuzzy reasoning.

Serial Number	Num	Time	Rule Results	Membership degree
1	few(N_0)	low(T_0)	low	Min(N_0, T_0)
2	few (N_0)	middle(T_1)	middle	Min(N_0, T_1)
3	few (N_0)	high (T_2)	middle	Min(N_0, T_2)
4	middle(N_1)	low (T_0)	low	Min(N_1, T_0)
5	middle(N_1)	middle (T_1)	middle	Min(N_1, T_1)
6	middle(N_1)	high (T_2)	high	Min(N_1, T_2)
7	many(N_2)	low (T_0)	low	Min(N_2, T_0)
8	many(N_2)	middle (T_1)	high	Min(N_2, T_1)
9	many(N_2)	high (T_2)	high	Min(N_2, T_2)

from 'Time' can trigger the relevant rules. For example, if the client' LDV is 'few' and RCT is 'high', the performance of the client is low. Fuzzy rule base is shown in Table 1.

The relationship between LDV and RCT is 'and', so the minimum value method is used to determine the strength of the rule. It can be observed that there are identical rule results, such as the first rule and the fourth rule. If several rules have the same result, the maximum membership degree of these rules is obtained using the maximum value method. The maximum value among the membership degrees of different rules is taken as the final determined output result, which is used to determine the level of the client.

5 A Method for Selecting Federated Learning Clients Based on Q-Learning

5.1 Design of Q-Table

We build a Q-matrix in the research, which is used to store all the learning performed by the clients through a series of actions and state transitions. The initial state of the Q-matrix represents a situation where it has no knowledge about the external environment, so all the values in the Q-matrix are set to 0. The rows represent states, which represent the current number of clients, ranging from 1 to 100. The columns represent actions, ranging from 1 to 100. However, only three actions are valid: increasing the number of clients by 1, decreasing the number of clients by 1, and keeping the number of clients unchanged. Other actions are considered invalid. For example, if the current state respectively is 10 clients, the corresponding valid actions are 9, 10 and 11.

5.2 Design of Reward Function

A simple WT is designed. The bandwidth is defined as 10 Mbps and converting it to bytes is 1310720 Bps. This means that 1310720 bytes can be transmitted in one second. A client network model size is calculated as model_ size = 2495610 bytes. As the number of clients increases, the utilization of bandwidth also changes accordingly. When clients' numbers are less than or equal to 10, the transmission time is defined as follows:

$$y = model\_size/(bandwidth * a) \tag{5}$$

The constant a represents the proportion of space in the bandwidth allocated for data transmission.

When clients' numbers are greater than10, the transmission time y is defined as follows:

$$y = model\_size/(bandwidth * (a - (n/10) * rate) \tag{6}$$

n represents the number of clients and *rate* represents a decrease rate for data transmission space.

The transmission time is defined as y. The accuracy of FL is acc. RCT is c. The reward function r is defined as follows:

$$r = 0.5 * (\frac{1}{100} * acc) + 0.2 * (\frac{1}{3.02} * y) + 0.3 * (\frac{1}{7360} * c) \tag{7}$$

5.3 An Iterative Process of Q-Learning

The iterative process of Q-learning begins by entering a different number of clients for each iteration. The FL model is then employed to obtain the corresponding FL accuracy, Round Completion Time (RCT), and transmission time for the given number of clients. These values are subsequently used as inputs to the reward function. Equation (8) represents the update rule for the Q-table in each iteration:

$$Q(s, a) = Q(s, a) + \alpha[r(s, a) + \gamma \max_A Q(s', a) - Q(s, a)] \tag{8}$$

α is the learning rate. $Q(s, a)$ represents the estimated value of the current state-action pair. $Q(s', a)$ represents the estimated value of the next state-action pair. $\max_A Q(s', a)$ selecting an action in action set A that maximizes $Q(s', a)$. γ represents the discount factor for the reward value. $r(s, a)$ represents the reward value immediately given by the environment when an action is executed in the current state.

6 Experimental Result

6.1 Experimental Dataset and Preprocessing

The dataset is the German Traffic Sign Recognition Benchmark. As shown in Fig. 6, the first row from left to right is speed limits of 20, 50, 60, 70 and 80. The second row from left to right is left turn, right turn, warning, no entry and STOP.

The client IDs for 1–100 are set to 0–99. If ID is 1, 31 or 61, the assigned LDV is greater than or equal to 600. If ID is divisible by both 3 and 5, the assigned LDV is greater than or equal to 200. If ID is divisible by 7, the assigned LDV is greater than or equal to 300. If ID is divisible by 8, the assigned LDV is greater than or equal to 400. As shown in Table 2, the assigned LDV is greater than or equal to 50 for IDs that do not meet the above conditions.

Fig. 6. Partial presentation of the GTSRB dataset.

Table 2. Client local data volume allocation results.

Local data volume	Clients' numbers
100	46
200	3
400	31
600	10
800	5
1000	4
1200	1

6.2 Metrics

We introduce a metric called the average resource computation time (ARCT) to assess the performance of the FL model. To define the $ARCT$, we first consider the total training completion time of the FL model, denoted as T_{FL}, and the number of clients participating in the training, represented by N. The ARCT is defined as follows:

$$ARCT = \frac{T_{FL}}{N} \tag{9}$$

6.3 A Method for Selecting Federated Learning Clients Based on Fuzzy Logic

The experimental results of the proposed scheme about selecting high-performance clients based on fuzzy logic are shown in Table 3.

To verify the effectiveness of the proposed scheme, we chose three baselines for comparison, namely, client selection method based on local data volume, client selection method based on resource computation time, and client selection method based on random sampling. The details are as follows.

If only considering clients' LDV, the clients with LDV in the range of 800 to 1200 are classified as high performance, and the clients with LDV in the range of 400 to 600 are classified as middle performance, and the clients with LDV in the range of 100 to 200 are classified as low performance. The results are shown in Table 4.

If only considering clients' RCT, the clients with RCT in the range of 64 to 67 are classified as high performance, the clients with RCT in the range of 67 to 71 are classified

Table 3. Client level classification (Client selection method based on fuzzy logic).

Client classification	Client ID
High	0, 1, 33, 65
Middle	4, 6, 8, 10, 12, 13, 14, 18, 25, 26, 29, 31, 36, 40,49, 50, 52, 54, 57, 60, 61, 63, 68, 69, 71, 78, 79, 82, 89
Low	2, 3, 5, 7, 9, 11, 15, 16, 17, 19, 20, 21, 22, 23, 24, 27, 28, 30, 32, 34, 35, 37, 38, 39, 41, 42, 43, 44, 45, 46, 47, 48, 51, 53, 55, 56, 58, 59, 62, 64, 66, 67, 70, 72, 73, 74, 75, 76, 77, 80, 81, 83, 84, 85, 86, 87, 88, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99

Table 4. Client level classification (Client selection method based on local data volume).

Client classification	Client ID
High	0, 1, 8, 16, 32, 33, 40, 65, 80, 88
Middle	3, 6, 7, 9, 12, 14, 15, 18, 21, 24, 28, 30, 35, 36, 39, 42, 45, 48, 49, 51, 54, 56, 57, 60, 63, 66, 69, 70, 72, 75, 77, 78, 81, 84, 87, 90, 91, 93, 98, 99
Low	2, 5, 10, 11, 13, 17, 19, 20, 22, 23, 25, 26, 29, 31, 34, 37, 41, 43, 44, 46, 47, 50, 52, 53, 55, 58, 59, 61, 62, 64, 67, 68, 71, 73, 74, 76, 79, 82, 83, 85, 86, 89, 92, 94, 95, 96, 97

as middle performance, and the clients with RCT in the range of 71 to 74 are classified as low performance. The results are shown in Table 5.

Table 5. Client level classification (Client selection method based on RCT).

Client classification	Client ID
High	1, 2, 5, 6, 10, 11, 13, 15, 16, 20, 22, 23, 24, 32, 33, 34, 38, 39, 41, 43, 44, 46, 47, 50, 51, 53, 54, 56, 57, 58, 59, 62, 63, 66, 68, 71, 74, 75, 76, 77, 78, 79, 80, 84, 85, 88, 90, 91, 92, 93, 96, 98, 99
Middle	0, 4, 7, 8, 9, 12, 14, 17, 18, 19, 21, 25, 26, 27, 28, 29, 30, 31, 35, 36, 40, 45, 48, 49, 52, 55, 60, 61, 64, 65, 67, 69, 72, 73, 81, 82, 83, 86, 87, 89, 94, 95, 97
Low	3, 37, 42, 70

The results of selecting clients using random sampling are shown in Table 6.

The evaluation metrics for this experiment are ARCT and the training data volume of the global model. The shorter ARCT of the selected clients in this group, the stronger the computational capability. The larger the training data volume of the global model of selected clients, the better the training performance of FL. The comprehensive analysis results of the experiments are shown in Fig. 7 and Fig. 8.

Table 6. Client level classification (Client selection method based on random sampling).

Client classification	Client ID
High	88, 19, 24, 34, 15, 74, 3, 22, 86, 60
Middle	45, 44, 53, 14, 52, 31, 61, 8, 46, 73, 30, 54, 32, 20, 21, 12, 40, 39, 27, 16, 18, 78, 26, 94, 17, 42, 89, 36, 99, 2, 85, 67, 84, 83, 1, 79, 82, 87, 49, 98, 23, 11, 77, 4, 7
Low	0, 5, 6, 9, 10, 13, 25, 28, 29, 33, 35, 37, 38, 41, 43, 47, 48, 50, 51, 55, 56, 57, 58, 59, 62, 63, 64, 65, 66, 68, 69, 70, 71, 72, 75, 76, 80, 81, 90, 91, 92, 93, 95, 96, 97

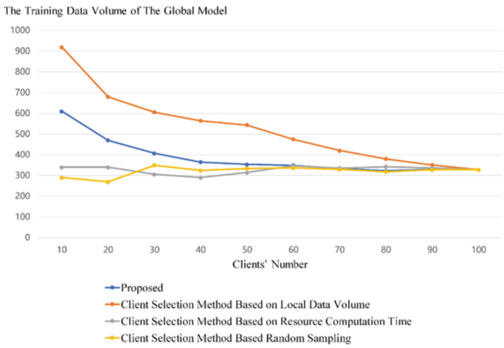

Fig. 7. Performance comparison of client level classification in global model training data volume for different methods.

There are a total of 100 clients, with 10, 20, 30 and up to 100 selected in sequence. Therefore, when all 100 clients are selected, the results of all methods are the same.

As shown in Fig. 7, client selection method based on LDV has a higher global model training data volume compared to the proposed method. However, as the number of clients increases, the difference between the two methods gradually diminishes until they become the same. The global model training data volume of client selection method based on RCT and client selection method based on random sampling is generally lower than that of the proposed method.

As shown in Fig. 8, the evaluation results indicate that the client selection method based on RCT has a higher ARCT compared to the proposed method. However, it is worth noting that as the number of clients increases, the difference between the two methods gradually diminishes, eventually converging to the same results. The abscissa of point A is x and the ordinate is y in Fig. 8. Specifically, when the number of clients is less than x, ARCT of client selection method based on LDV is lower than the proposed method. When the range of clients' number is x ~ 50, ARCT of client selection method based on LDV is higher than the proposed method. When the clients' number is greater than 50, both methods become similar and eventually converge to the same results. When clients' number is less than 40, ARCT of client selection method based on random sampling is lower than the proposed method. When the clients' number is greater than 40, both methods become similar and eventually converge to the same results. These findings highlight the capability of the proposed method to effectively optimize client selection in the context of federated learning in multi-robot collaborative systems.

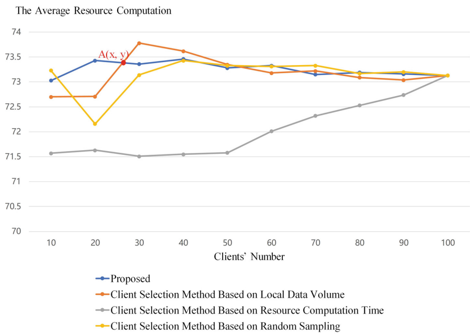

Fig. 8. Performance comparison of client level classification in ARCT for different methods.

6.4 A Method for Selecting Federated Learning Clients Based on Q-Learning

In a collaborative system of multiple mobile robots (clients), there are several robots collaborating to transport goods. Each robot has varying volumes of local data consisting of traffic sign images. The communication bandwidth between the robots is 10 Mbps. However, when the robots' numbers are greater than 10, the data transmission portion of the bandwidth decreases at a rate of 0.02. Due to limited network resources, it is not necessary to select all mobile robots for communication. In the first scenario, there are a total of 30 robots working Collaboratively, the optimal number of robots determined is 15 according to the proposed method. In the second scenario, there are a total of 50 robots working Collaboratively, the optimal number of robots determined is 40 according to the proposed method.

The evaluation metrics for this experiment are ACC and ARCT. As the number of clients increases, the changes in model accuracy and ARCT are respectively shown in Fig. 9 and Fig. 10.

The experimental results obtained by the proposed method are 15 robots in the first scenario. From ACC perspective, the result is a local optimal solution in between 5 and 30. From ARCT perspective, the result is a local optimal solution between 16 and 30.

The experimental results obtained by the proposed method are 40 robots in the second scenario. From ACC perspective, the ACC of the result is low. From ARCT perspective, the result is a local optimal solution between 41 and 50.

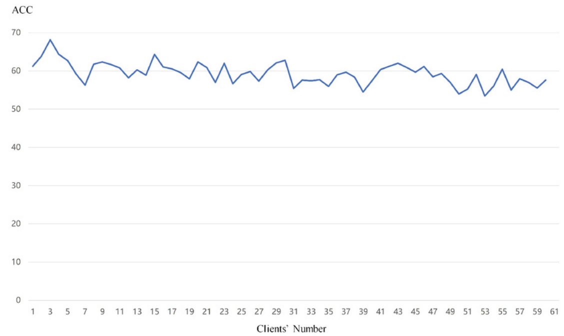

Fig. 9. The relationship between clients' numbers and ACC.

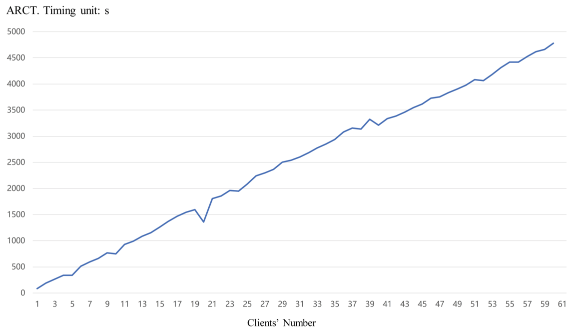

Fig. 10. The relationship between clients' numbers and ARCT.

7 Conclusion

In this paper, we propose a client selection method for federated learning in multi-robots collaborative systems. The proposed method combines fuzzy logic and Q-learning to address the challenges of selecting high-performance clients and determining the appropriate number of clients for FL. By employing fuzzy logic, the method effectively identifies and selects high-performance clients based on their data volume and computational capabilities. This selection process improves communication efficiency and overall work performance within the collaborative system. Moreover, the integration of Q-learning enables the determination of the optimal number of clients for FL. By considering factors such as network constraints, computational resource capabilities, and FL model accuracy, the method achieves a balance between the number of clients and their performance. This approach ensures efficient resource utilization and enhances the overall effectiveness of FL. The effectiveness of the proposed method is verified through experimental results and comparative analysis. In this research, the number of robots selected based on different scenarios is considered to be a local optimal solution. Consequently, our subsequent efforts will be focused on further optimizing it towards a global optimum solution.

Acknowledgments. This research was supported in part by the Inner Mongolia Science and Technology Key Project No. 2021GG0218, ROIS NII Open Collaborative Research 23S0601, and in part by JSPS KAKENHI Grant No. 21H03424.

References

1. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Proceedings of the Annual Conference on Neural Information Processing Systems, pp. 1097–1105 (2012)
2. Sun, C., Shrivastava, A., Singh, S., Gupta, A.: Revisiting unreasonable effectiveness of data in deep learning era. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 843–852 (2017)
3. Yadav, R., Zhang, W., Kaiwartya, O., Song, H., Yu, S.: Energy-latency tradeoff for dynamic computation offloading in vehicular fog computing. IEEE Trans. Veh. Technol. **69**(12), 14198–14211 (2020)
4. McMahan, H.B., Ramage, D., Talwar, K., et al.: Learning differentially private recurrent language models. arXiv preprint arXiv:1710.06963 (2017)
5. Apple Differential Privacy Team. Learning with privacy at scale. In Apple Machine Learning Journal (2017)
6. Hartmann, F., Suh, S., Komarzewski, A., et al.: Federated learning for ranking browser history suggestions. arXiv preprint arXiv:1911.11807 (2019)
7. Zhu, H., Xu, J., Liu, S., et al.: Federated learning on non-IID data: a survey. Neurocomputing **465**, 371–390 (2021)
8. Zhao, Y., Li, M., Lai, L., et al.: Federated learning with non-IID data. arXiv preprint arXiv:1806.00582 (2018)
9. IEEE J. Sel. Areas Commun. **41**(4), 915–928 (2023). https://doi.org/10.1109/JSAC.2023.3242720
10. Zhang, W., Wang, X., Zhou, P., Wu, W., Zhang, X.: Client selection for federated learning with non-IID data in mobile edge computing. IEEE Access **9**, 24462–24474 (2021)
11. Qu, Z., Duan, R., Chen, L., Xu, J., Lu, Z., Liu, Y.: Context-Aware online client selection for hierarchical federated learning. IEEE Trans. Parallel Distrib. Syst. **33**(12), 4353–4367 (2022)
12. Huang, T., Lin, W., Shen, L., Li, K., Zomaya, A.Y.: Stochastic client selection for federated learning with volatile clients. IEEE Internet Things J. **9**(20), 20055–20070 (2022)
13. Asad, M., Moustafa, A., Rabhi, F.A., Aslam, M.: THF: 3-way hierarchical framework for efficient client selection and resource management in federated learning. IEEE Internet Things J. **9**(13), 11085–11097 (2022)
14. Shi, F., Hu, C., Lin, W., Fan, L., Huang, T., Wu, W.: VFedCS: optimizing client selection for volatile federated learning. IEEE Internet Things J. **9**(24), 24995–25010 (2022)