



Federated Learning Based User Scheduling for Real-Time Multimedia Tasks in Edge Devices

Wenkan Wen¹, Yiwen Liu^{1,2,3}(✉), Yanxia Gao^{1,2,3}, Zhirong Zhu¹, Yuanquan Shi^{1,2,3},
and Xiaoning Peng^{1,2,3}

¹ School of Computer and Artificial Intelligence, Huaihua University, Huaihua 418000, China
lyw@hhtc.edu.cn

² Key Laboratory of Wuling-Mountain Health Big Data Intelligent Processing and Application
in Hunan Province Universities, Huaihua 418000, Hunan, People's Republic of China

³ Key Laboratory of Intelligent Control Technology for Wuling-Mountain Ecological
Agriculture in Hunan Province, Huaihua 418000, Hunan, People's Republic of China

Abstract. Edge networks are highly volatile and the quality of device communication and computational resources change not only over time but also according to the movement of users. Current federation learning suffers from poor device network state and failure of devices to upload models in a timely manner. To address these problems, an intelligent scheduling mechanism that uses the predicted device state based on device information to select the appropriate device for federated learning is proposed in this paper. By focusing on information such as communication quality, computational resources, and location information, the information of edge devices is collected to analyze and predict the device network and computing resources to further analyze the state of devices in depth. Experiments are conducted on real datasets, and the experimental results show that the proposed scheduling method can make the global model fit faster than without the algorithm, which significantly improves the training efficiency of federated learning.

Keywords: Federated Learning · User Scheduling · Mobile Computing · Machine learning

1 Introduction

1.1 Background

With the development of technology and the increasing improvement of people's living standard, electronic products such as mobile communication devices and wearable devices are becoming more and more popular, and a large amount of data is generated in the use of these devices by users. At the same time, artificial intelligence is also developing rapidly and a dilemma has arisen: on the one hand, most of the application models of artificial intelligence technology need a large number of rich sources of real user data as the basis for training. On the other hand, the importance of user privacy and data

security has been increasing in recent years [1]. In addition, the lack of an effective interoperability and collaboration among commercial companies applying AI technologies has resulted in the potential data value not being fully exploited and applied. Federated Learning (FL) [2] can be a good solution to the above dilemma as a technical solution that enables multiple devices distributed in multiple regions to collaboratively train AI models using local data while ensuring privacy. Specifically, federal learning differs from the traditional learning model in that the server does not need direct access to the user's local data, but rather allows the device to train its own data locally. In each round of training, the local client can download the current global model from the server, update the model by training on local data, and upload the updated model to the server. Then, the server will integrate all the uploaded models with the updated global model through the algorithm. In this way, local data from different clients do not exchange with each other, and the transmission information of model updates can be further protected by encryption algorithms [3]. For example, an approach based on model summation averaging enables Google's G Board input method to effectively predict the next character the user wishes to type [4].

Google proposed a scalable federation learning system based on mobile devices in 2019 and defined three phases of federation learning. First is the Device Selection Phase, which requires the central Parameter Server (PS) of federation learning to select a set of devices. Clients, for model training. Next is the Local-training Phase, which requires the devices selected in the previous phase to be trained locally. Finally, the Model Synchronization Phase, also known as the Reporting Phase, requires the devices that have completed local training to upload the updated local models or gradients to the parameter server, and the parameter server completes the global model aggregation. After the above three phases are completed, the next device selection phase is entered again, and the cycle is repeated until the model performance meets the expected requirements.

1.2 Challenge

One of the keys to the quality of federation learning training for edge networks lies in the devices involved in model training and synchronization, and there are two main factors in edge networks that can greatly affect the final global model merit.

- (1) Mobile devices may fluctuate in network communication quality due to their irregular locations and frequent movements. This will easily make it difficult for the device to communicate with the base station or the parameter server, thus affecting the effectiveness of federal learning.
- (2) The computing power of the device is influenced by the tasks performed by the device itself, so the computing power of the device is a dynamic situation, and the high or low computing power of the device will directly affect the overall federal learning training efficiency and the final recognition ability of the model.

The instability of mobile devices is high, and the network communication of devices in edge networks is always in constant change. Therefore, a good device scheduling method can steadily and effectively improve the training efficiency and model stability of federation learning [5–9]. In this paper, we propose an intelligent scheduling mechanism

that uses mobile device state information to predict the device state and then selects the device with better performance for federation learning. The network and computing resources of the predicted devices are analyzed by collecting information from edge devices to further analyze the state of the devices. A machine learning approach is used to model the real mobile devices based on their state data, and the information provided by the current devices such as computational performance, network state, and location information is used to make a comprehensive decision to ensure that the most stable device is selected in each round of federal learning device selection.

2 Related Review

Distributed machine learning has received a great deal of scholarly attention and has yielded numerous results [10, 11], and a number of large-scale cloud-based distributed systems have been put into use. Many systems support multiple data processing schemes, including model parallelism and data parallelism [12]. Distributed edge intelligence is a disruptive area of research that enables machine learning and deep learning (ML/DL) algorithms to be executed where the data is generated. Machine learning and deep learning (ML/DL) algorithms are executed close to where the data is generated [13]. Current distributed learning considers devices with high storage and computational power and stable network connections, such as computers, micro servers, etc. Not much attention has been paid to devices with weaker performance and susceptible to fluctuations such as cell phones, wearable devices, etc.

Federated Learning, first proposed by Google, aims to enable multiple parties to work together on machine learning tasks while protecting data privacy. With the enactment of the European Union's General Data Protection Regulation (GDPR), large Internet companies are no longer free to collect user data for machine learning. As a result, Google proposed in 2016 a new policy to protect user data. Therefore, Google proposed a federal learning approach for Android phone users in 2016, which can train a federated model while ensuring that user data does not leave the local device. The literature [14] extends Google's federated learning concept with a more general definition and classifies federated learning into horizontal federated learning, vertical federated learning and federated migration learning for different ways of slicing data [15]. In either scenario, there are several common challenges [14].

In the case of two datasets with more repetition of user's feature dimensions and less overlap of users, we slice the dataset horizontally and take out the data with the same but not identical features for training. Longitudinal federation learning refers to the case where the users overlap more and their feature dimensions overlap less in the two datasets, the dataset is cut vertically and the data with the same but not all the same features are taken out from the two datasets for training. Federal transfer learning [16] refers to the case where there is little overlap in both user dimensions and user feature dimensions in both datasets. In this case migration learning techniques are used to overcome the problem of data and label imbalance.

L. U. Khan [17] considered the need for a new protocol to select a set of local devices with sufficient resources to provide the standard. The selection criteria for the devices must include persistent backup power, sufficient memory, accurate data and high

processing power [18–22]. The article [23] argues that in federated learning, the participating users are heterogeneous in terms of their computational power and willingness to participate; communication cost is one of the bottlenecks in federated learning, and it is very natural and reasonable to introduce mobile edge computing into federated learning to leverage communication and computational costs. When using mobile edge computing in federation learning, the latency from the client to the edge server is considered lower than the link between the client and the cloud, so the edge server is introduced for local parameter aggregation, and considering the heterogeneity of the clients, various resources should be carefully allocated to improve the training efficiency, such as client grouping, joint radio and computational resource management, and adaptive aggregation. The paper [24] proposes a dynamic sampling strategy that first uses a higher sampling rate for dynamic sampling and then decreases the sampling rate at each communication. First to speed up the beginning phase of convergence, so by involving more users in the model training. When training reaches a certain level, the number of users involved in training is dynamically reduced to save communication costs. The initial cost of federal learning is high, but after a few rounds of training, the number of users selected for model training decreases rapidly. The rate of reduction can be freely chosen to suit the situation in different environments. The article [25] shows that different users have different network speeds and communication methods. And the distribution of the data owned by the clients is unbalanced during the training process. If all users are allowed to participate in the training process of joint learning, then there will be participants who are behind in the iterations. If some users do not respond for a long time, the whole system may not be able to complete the joint training. Therefore, considering how to select the right devices to participate in federated learning is also a major urgent problem.

The above research results show that federated learning for edge computing has certain requirements on the computational power, network stability, and available memory of the devices. Therefore, it is important to collect the information of edge devices and consider the device performance to select the appropriate devices for federated learning.

3 Data and Algorithm

The experimental environment is a mobile edge communication (MEC) environment consisting of a cloud server, several groups of base stations (including macro and micro base stations) and several mobile candidates, represented by the set of $N = \{1, 2, 3, \dots, N\}$. Each device has a certain amount of available CPU and memory resources, as well as a network connection with a certain bandwidth already, which are set to be determined by the available computing power and the upload/download rate of the device, respectively. A candidate device's available CPU resources and memory size represent its currently available computational resources, while its network connectivity determines whether it can download a global model or upload a locally trained model at a reasonable time.

As shown in Fig. 1, the learning process consists of collaboration between the cloud server, macro base stations, micro base stations and mobile terminals to collaborate on model training. The cloud server broadcasts the global model to the macro base station, which in turn broadcasts it to all micro base stations, which in turn forwards the received global model to the edge devices. After training, the edge devices upload

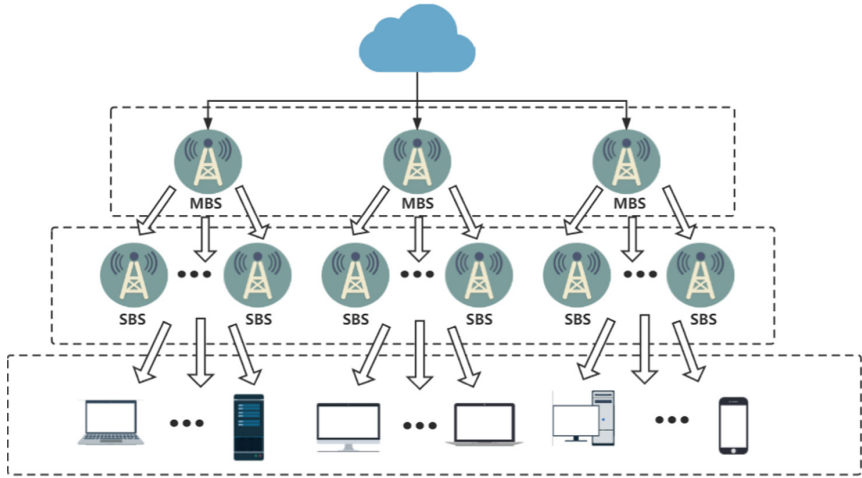


Fig. 1. An edge federation learning environment consists of four components: cloud servers, macro base stations, micro base stations, and edge device.

the generated local models to the associated micro-base stations. The micro-base station receives all the updated local models, performs a weighted average of the local models, and then sends the results to the macro-base station. Finally, the macro base station then pushes the models to the cloud server. Considering the large number of edge devices, the performance and communication capability are inconsistent. Each micro-base station has an uncertain update time for each round, so an asynchronous update strategy will be used between the macro-base station and the micro-base station (Table 1).

Table 1. Notations and their description in this paper

Symbols	Description
N	Number of mobile devices
S	Number of base stations
n	Mobile Devices
s	Micro Base Station
B_e	Device remaining battery
D_n	Dataset to be processed
P_n	Calculated power consumption
L_n	Device training rounds

(continued)

Table 1. (continued)

Symbols	Description
B_c	Power required for training
A_n	Device Performance
T_t	Training time
N_u	Network upload rate
N_d	Network download rate
N_n	Network Quality

3.1 Data

The dataset in this paper is from the article [26], the device generates a large amount of data during operation, by collecting this information the following dimensional information can be obtained: Latitude, Longitude, Date and Time, Available Memory, Upload Rate, Download Rate, Battery Whether the device is able to pass the completed training parameters to the base station at a reasonable time will be used as a label for the suitability of the device (Table 2).

Table 2. Dataset information table

Num	Item	Description
1	Latitude	Device current latitude coordinates
2	Longitude	Device current longitude coordinates
3	Date and Time	Current date and time
4	Available Memory	Device available memory
5	Upload Rate	Network upload rate
6	Download Rate	Network download rate
7	Battery Remain	Device remaining battery
8	Max CPU Frequency	CPU Max Frequency
9	Min CPU Frequency	CPU Min Frequency
10	Current CPU Frequency	CPU Current Frequency

3.2 Algorithm

In edge computing environments, where performance varies greatly between devices, many factors can affect the quality of the final model of federation learning. The power of the device, network status, and location are all information to focus on.

(1) **Device residual power constraint**

The power of the device is affected by the power consumption of the device, the size of the data set to be processed and the number of training rounds required. In order to ensure that the device can have enough power to complete the training and parameter transfer, the selected device needs to be constrained. The formula is shown below:

$$B_c = D_n * P_n * L_n \quad (1)$$

$$|B_e - B_c| > \lambda B_c \quad (2)$$

Formula (1) can calculate the approximate power demand for this round of training, formula (2) is to determine the difference between the remaining power of the device and the power required for training needs to ensure that it is at least twice as large as the power required for training before it can be selected as a training device, otherwise the device will not be selected as a training device.

(2) **Device network state constraints**

In real scenarios the network bandwidth of mobile devices may be affected by users resulting in slow or even failed transmission. And the bandwidth that the edge devices can provide is small [27], so if the parameters are disconnected during the transmission process, it will affect the performance of the federation learning global model, so the selected devices need to be constrained. The formula is shown as follows:

$$N_n = \frac{U_n}{\sum_{i=1}^N \frac{U_i}{N}} + \frac{D_n}{\sum_{i=1}^N \frac{D_i}{N}} \quad (3)$$

$\sum_{i=1}^N \frac{U_i}{N}$ and $\sum_{i=1}^N \frac{D_i}{N}$ is the average of the current network upload and download rates of all candidate devices, The sum of the network upload rate U_n of the currently selected device and the network download rate D_n of the currently selected device and the corresponding ratio of all candidate devices is the current device network quality N_n . The device network quality N_n is a greater than zero constant, the larger the value the worse the quality of the currently selected device network and vice versa.

(3) **Device Computational Performance Constraints**

The processing performance of mobile devices and the time required for transmission in real-world scenarios are the key concerns. Edge devices are usually weak in computation, and if the training time is long, it will affect the performance of the federated learning global model, so constraints need to be placed on the selected devices. The formula is shown as follows:

$$A_n = \frac{(P_{cur} + P_{max})}{P_{max}} \times P_{min} \quad (4)$$

$$T_t = \frac{D_n}{A_n} * L_n \quad (5)$$

The relative performance of the mobile device can be calculated by formula (4), and the actual computing performance of the device can be judged by the relative performance. Combined with Eq. (5) will calculate the approximate training time, the smaller the training time the better the performance will be, the higher the probability of being selected, and vice versa.

(4) **Device position offset constraint**

Mobile devices usually do not stay fixed in one location and may continuously switch sites [28], in which process it can cause the base station assigned to the mobile device task to have difficulty receiving mobile device information or the mobile device is out of the signal range of the base station. So it is necessary to make restrictions on the mobile information of the device.

$$\text{haversin}\left(\frac{d}{R}\right) = \text{haversin}(\varphi_2 - \varphi_1) + \cos(\varphi_1) \cos(\varphi_2) \text{haversin}(\Delta\lambda) \quad (6)$$

$$\text{haversin}(\theta) = \sin^2(\theta/2) = (1 - \cos(\theta))/2 \quad (7)$$

$$D_m = \text{haversin}(|L_{\tilde{x}} - L_x|) + \cos(L_x) \cos(L_{\tilde{x}}) \text{haversin}(|L_{\tilde{y}} - L_y|) \quad (8)$$

$$O(S_a, D_m) = \begin{cases} 1 & |S_a - D_m| < 0 \\ 0 & 0 \leq |S_a - D_m| \end{cases} \quad (9)$$

$$O_c = \sum_{i=1}^t O(S_a, D_m) \quad (10)$$

The device latitude and longitude information is obtained and the Haversine algorithm [29] (Formula (6), (7), and (8)) is used to calculate the distance between the previous moment and the current moment of the device movement. After that, formula (9) is used to determine whether the device exceeds the coverage range of the base station, and the departure from the coverage range of the base station will cause the device communication to be affected. Formula (10) records the number of times the device switches base stations to determine whether the device is stable or not. The lower the number of times the device switches base stations, the higher the quality of the device, and vice versa (Fig. 2).

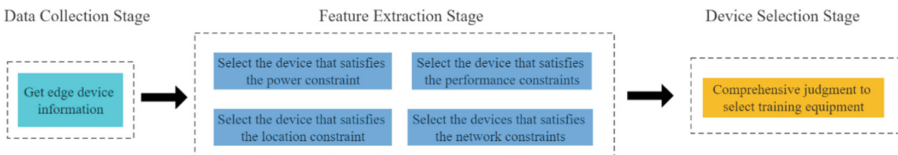


Fig. 2. The scheduling process can be divided into three stages: data collection stage, feature extraction stage, and device selection stage

4 Experiments

This experiment will use several machine learning models for comparative validation experiments to better verify the performance of the intelligent scheduling algorithm in different models.

Native Bayesian classification is an NB algorithm proposed by Maron and Kuhns based on Bayesian theory for classification based on probabilistic principles. The Native Bayesian algorithm is a classification method based on probabilistic statistics and is often used as an evaluation criterion for text classification. After obtaining the prior probability and conditional probability of the sample data, the algorithm can find the posterior probability according to the Bayesian formula, the probability that the sample corresponds to different categories, and the category with the largest posterior probability is the predicted category of the sample corresponding to the algorithm [30].

Support vector machines, as a powerful classification and regression method, have achieved very remarkable research results and are widely used in many fields. Support vector machines are based on the statistical principle of structural risk minimization and obtain the optimal division hyperplane between positive and negative classes by maximizing the soft interval [31].

Decision tree (DT) algorithms can be used in classification and regression problems and have been widely used in knowledge discovery, pattern recognition, and other fields. Compared with other machine learning models, decision trees have the advantages of high generalization ability, fast construction, few parameters, and easy understanding. To this day, decision trees are still one of the research hotspots in the field of machine learning, with specific research focusing on tree construction methods, node partitioning methods and partitioning criteria decision tree integration models, combinations with other classifiers, and other directions [32].

The random forest (RF) algorithm is concerned with integrated learning of decision trees and was proposed by Breiman in 2001. This algorithm requires simulation and iteration and is classified as a method in machine learning. Later, some scholars borrowed the proposed random decision forests approach to combine classification trees into random forests, i.e., to randomize the use of variables (columns) and data (rows) to generate many classification trees, and then aggregate the results of classification trees. Later Breiman published the algorithm of random forest designed by him and Cutler in the *Journal of Machine Learning*, and this article was cited extensively and became a milestone in the field of machine learning [33].

4.1 Evaluation Metric

The experimental procedure will use accuracy (Precision, P) (as in Eq. 10), recall (Recall, R) (as in Eq. 11), the overall evaluation metric (F1-score) (as in Eq. 12), and the ROC curve to evaluate the performance of the classifier algorithm (TP: correct rate; FP: false alarm rate; FN: missed alarm rate). F1-score is widely used in information retrieval, machine learning, sentiment analysis, and other fields involving binary classification.

The ROC curve provides a global assessment of the classifier's accuracy, and it contains all possible decision thresholds without a specific decision threshold. In addition, ROC curves do not depend on the scale of the test results, i.e., monotonic transformations

of the classification results can maintain invariance, thus, in most practical problems, the problem of uneven sample distribution and inconsistent cost of misclassification may affect the evaluation results, and ROC curves are a good solution to this challenge. In terms of performance diagnosis, ROC curves are free distribution assumptions in the full sense of the word, without the need to use the parametric form of the assumed class distribution, thus ensuring the wide applicability of ROC curves [34].

$$Precision = \frac{TP}{TP + FP} \quad (11)$$

$$Recall = \frac{TP}{TP + FN} \quad (12)$$

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (13)$$

4.2 Result

Precision (P), which is specific to our prediction results, indicates how many of the positive predictions are actually positive in the original sample. Recall (R), which is specific to our original sample, indicates how many of the positive cases in the original sample were predicted correctly, the percentage of true and accurate positive predictions among the positive cases in the original sample. F1-score is a composite precision and recall metric, and F1 value is the summed mean of precision and recall. According to the above Eqs. (11), (12) and (13), the P-value, R-value and F1-score before and after comparing q using the scheduling algorithm are shown in Table 3.

Table 3. Model Performance Table

Model	Accuracy	Precision	Recall	F1-score
Native Bayesian	78.89%	0.72	0.74	0.73
SVM	83.73%	0.79	0.77	0.78
Decision Tree	91.78%	0.88	0.90	0.89
Random Forest	92.31%	0.89	0.91	0.90

As seen in Table 3, the models all achieved good results, with Random Forest achieving the highest F1-score score, Decision Tree also achieving better prediction results, SVM the next best, and Native Bayes the worst. Among them, random forest obtained 92.31% accuracy, which shows that the selected features can effectively represent the state of the device and can effectively and stably predict the state of the device selected with better performance to participate in the federal learning.

Figure 3 shows that the model still has good performance in predicting the suitable candidate devices even at 0.1. It can be shown that the selected features can achieve a

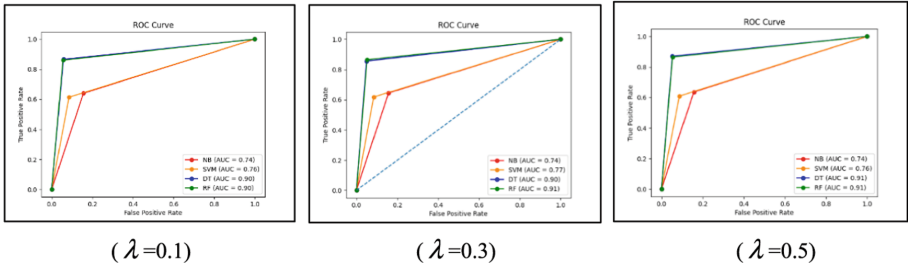


Fig. 3. It can be found that the model still predicts the device that should be selected well for different λ values, indicating that the features extracted by the algorithm are effective.

more satisfactory accuracy rate, which lays the foundation for the subsequent experiments and analysis. Afterwards, the RESNET50 model will be tested using the classical MNIST dataset to verify the effectiveness of the intelligent scheduling algorithm in federal learning by comparing the convergence speed of the global model before and after 1000 rounds of federal learning training using the intelligent scheduling algorithm.

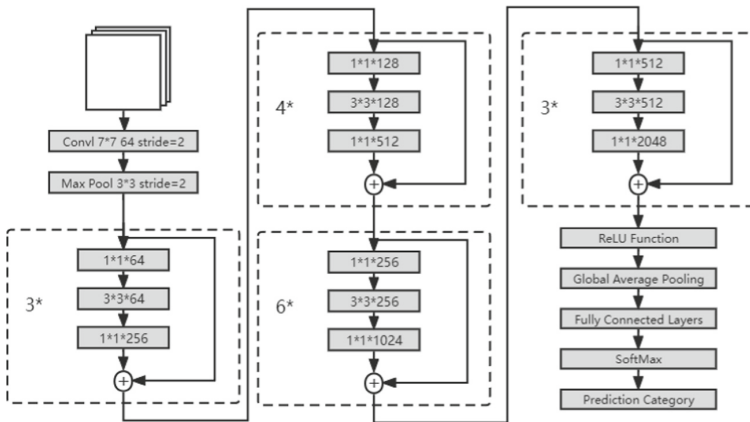


Fig. 4. This is a diagram of a classic Resnet50 network structure with 49 convolutional layers and one fully connected layer

Figure 4 shows the network structure of Resnet50 used for the experiments by combining the data set mobile device information with a randomly generated unique ID to derive the data set. We randomly select 1000 devices for the simulation, and each device randomly selects a default base station to communicate with. Afterwards, 80% of the data from MNIST is used as the training set for federal learning, and the remaining 20% is used as the test set to verify the performance of the trained global model.

A comparison of the convergence performance using the smart scheduling algorithm with and without is shown in Fig. 5. It is clearly observed from the figure that the model using the smart scheduling method has a faster convergence rate and obtains high accuracy compared to the model without smart scheduling. This is a good indication of the

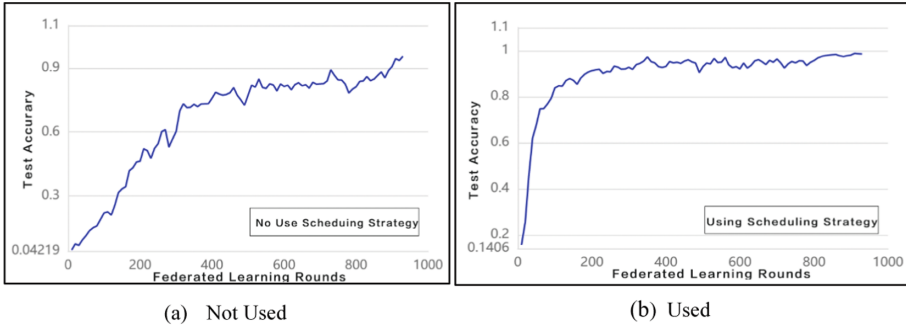


Fig. 5. It is obvious that the model converges significantly faster with the scheduling algorithm than without it

advantage of the smart scheduling algorithm with more stable convergence. Therefore, it is a feasible and effective solution to collect and analyze the device information for intelligent scheduling to select the appropriate devices for federal learning.

5 Conclusion

This paper focuses on a federated learning approach to device scheduling, and judges the capabilities of devices comprehensively by focusing on three important features: communication quality, computational resources, and location information. And considering that edge networks have a large turbulence, the device communication quality and computational resources not only change over time but also according to the movement of users. The experiments are conducted by invoking the machine learning model for federal learning device selection and using Resnet50 to verify the effectiveness of the algorithm.

Acknowledgment. This work was supported in part by the Scientific research projects funded by the Department of education of Hunan Province (No. 22C0497), the Huaihua University Double First-Class initiative Applied Characteristic Discipline of Control Science and Engineering (No. ZNKZN2021-10), the National Natural Science Foundation of China (No. 62172182), the Hunan Provincial Natural Science Foundation of China (No. 2020JJ4490), the Project of Hunan Provincial Social Science Foundation (No. 21JD046), the Huaihua University Project (No. HHUY2019-25), the Philosophy and Social Science Achievement Evaluation Committee of Huaihua (No. HSP2022YB40) and the Science and Technology Innovation 2030 Special Project Sub-Topics (No. 2018AAA0102100).

References

1. Su, W., Liu, D., Zhang, T., Jiang, H.: Towards device independent eavesdropping on telephone conversations with built-in accelerometer. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **5**(4), 1–29 (2021)

2. Bonawitz, K., et al.: Towards federated learning at scale: system design. *Proc. Mach. Learn. Syst.* **1**, 374–388 (2019)
3. Aono, Y., Hayashi, T., Wang, L., Moriai, S.: Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Trans. Inf. Forensics Secur.* **13**(5), 1333–1345 (2017). <https://doi.org/10.1109/TIFS.2017.2787987>
4. Hard, A., et al.: Federated learning for mobile keyboard prediction. arXiv preprint [arXiv:1811.03604](https://arxiv.org/abs/1811.03604) (2018)
5. Zeng, F., Li, Q., Xiao, Z., Havyarimana, V., Bai, J.: A price-based optimization strategy of power control and resource allocation in full-duplex heterogeneous macrocell-femtocell networks. *IEEE Access* **6**, 42004–42013 (2018)
6. Zeng, F., et al.: Resource allocation and trajectory optimization for QoE provisioning in energy-efficient UAV-enabled wireless networks. *IEEE Trans. Veh. Technol.* **69**(7), 7634–7647 (2020)
7. Ali, T.A.A., Xiao, Z., Sun, J., Mirjalili, S., Havyarimana, V., Jiang, H.: Optimal design of IIR wideband digital differentiators and integrators using salp swarm algorithm. *Knowl.-Based Syst.* **182**, 104834 (2019)
8. Jiang, H., Cao, H., Liu, D., Xiong, J., Cao, Z.: SmileAuth: using dental edge biometrics for user authentication on smartphones. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **4**(3), 1–24 (2020)
9. Liu, D., Cao, Z., Hou, M., Rong, H., Jiang, H.: Pushing the limits of transmission concurrency for low power wireless networks. *ACM Trans. Sens. Netw.* **16**(4), 1–29 (2020)
10. Xiao, Z., et al.: Toward accurate vehicle state estimation under non-Gaussian noises. *IEEE Internet Things J.* **6**(6), 10652–10664 (2019)
11. Hu, J., et al.: BlinkRadar: non-intrusive driver eye-blink detection with UWB radar. In: 2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS), pp. 1040–1050. IEEE (2022)
12. Lu, X., Liao, Y., Lio, P., Pan, H.: An asynchronous federated learning mechanism for edge network computing. *J. Comput. Res. Dev.* **57**(12), 2571–2582 (2020)
13. Filho, C.P., et al.: A systematic literature review on distributed machine learning in edge computing. *Sensors* **22**(7), 2665 (2022). <https://doi.org/10.3390/s22072665>
14. Yang, Q., Liu, Y., Chen, T., Tong, Y.: Federated machine learning: concept and applications. *ACM Trans. Intell. Syst. Technol.* **10**(2), 1–19 (2019). <https://doi.org/10.1145/3298981>
15. Zhang, P.-C., Wei, X.-M., Jin, H.-Y.: Dynamic QoS optimization method based on federal learning in mobile edge computing. *Chin. J. Comput.* **44**(12), 2431–2446 (2021)
16. Liu, Y., Kang, Y., Xing, C., Chen, T., Yang, Q.: A secure federated transfer learning framework. *IEEE Intell. Syst.* **35**(4), 70–82 (2020). <https://doi.org/10.1109/MIS.2020.2988525>
17. Khan, L.U., et al.: Federated learning for edge networks: resource optimization and incentive mechanism. *IEEE Commun. Mag.* **58**(10), 88–93 (2020). <https://doi.org/10.1109/MCOM.001.1900649>
18. Liu, D., Cao, Z., He, Y., Ji, X., Hou, M., Jiang, H.: Exploiting concurrency for opportunistic forwarding in duty-cycled IoT networks. *ACM Trans. Sens. Netw.* **15**(3), 1–33 (2019)
19. Hu, Z., Zeng, F., Xiao, Z., Fu, B., Jiang, H., Chen, H.: Computation efficiency maximization and QoE-provisioning in UAV-enabled MEC communication systems. *IEEE Trans. Netw. Sci. Eng.* **8**(2), 1630–1645 (2021)
20. Jiang, H., Dai, X., Xiao, Z., Iyengar, A.K.: Joint task offloading and resource allocation for energy-constrained mobile edge computing. *IEEE Trans. Mob. Comput.* (2022). <https://doi.org/10.1109/TMC.2022.3150432>
21. Jiang, H., Xiao, Z., Li, Z., Xu, J., Zeng, F., Wang, D.: An energy-efficient framework for internet of things underlying heterogeneous small cell networks. *IEEE Trans. Mob. Comput.* **21**(1), 31–43 (2020)

22. Liu, D., Hou, M., Cao, Z., He, Y., Ji, X., Zheng, X.: COF: exploiting concurrency for low power opportunistic forwarding. In: 2015 IEEE 23rd International Conference on Network Protocols (ICNP), pp. 32–42. IEEE (2015)
23. Qin, Z., Li, G.Y., Ye, H.: Federated learning and wireless communications. *IEEE Wireless Commun.* **28**(5), 134–140 (2021). <https://doi.org/10.1109/MWC.011.2000501>
24. Ji, S., Jiang, W., Walid, A., Li, X.: Dynamic sampling and selective masking for communication-efficient federated learning. *IEEE Intell. Syst.* **37**(2), 27–34 (2022). <https://doi.org/10.1109/MIS.2021.3114610>
25. Alferaidi, A., Yadav, K., Alharbi, Y., Viriyasitavat, W., Kautish, S., Dhiman, G.: Federated learning algorithms to optimize the client and cost selections. *Math. Probl. Eng.* **2022**, 8514562 (2022). <https://doi.org/10.1155/2022/8514562>
26. Huang, H., Li, R., Liu, J., Zhou, S., Lin, K., Zheng, Z.: ContextFL: context-aware federated learning by estimating the training and reporting phases of mobile clients. In: 2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS), pp. 570–580. IEEE (2022)
27. Liu, D., Wu, X., Cao, Z., Liu, M., Li, Y., Hou, M.: CD-MAC: a contention detectable MAC for low duty-cycled wireless sensor networks. In: 2015 12th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), pp. 37–45. IEEE (2015)
28. Qian, C., Liu, D., Jiang, H.: Harmonizing energy efficiency and QoE for brightness scaling-based mobile video streaming. In: 2022 IEEE/ACM 30th International Symposium on Quality of Service (IWQoS), pp. 1–10. IEEE (2022)
29. Inman, J.: Navigation and Nautical Astronomy: For the Use of British Seamen 3. In: Woodward, W.C., Rivington, J. (eds.) London, UK 1835 (1821)
30. Li, X., Yang, Z., Ren, J.: Improved naive bayes algorithm based on dual feature selection of mutual information and hierarchical clustering measurement & control technology **41**(02), 36–40+69 (2022). <https://doi.org/10.19708/j.ckjs.2022.02.005>
31. Wang, X., Dong, Y., Yu, Q., Geng, N.: Review of structural support vector machines. *Comput. Eng. Appl.* **56**(17), 24–32 (2020). (in Chinese)
32. Liu, Z., Chu, N.: A weighted clustering splitting decision tree algorithm. *Telecommun. Eng.* **60**(11), 1354–1360 (2020)
33. Li, X.: Using “random forest” for classification and regression. *Chin. J. Appl. Entomol.* **50**(4), 1190–1197 (2013)
34. Wang, Y., Zhu, H., Xu, W.: A review on ROC curve and analysis. *J. Guangdong Univ. Technol.* **38**(01), 46–53 (2021)