

Research Article

WCL: Client Selection in Federated Learning with a Combination of Model Weight Divergence and Client Training Loss for Internet Traffic Classification

Yingya Guo ^{1,2,3} Kai Huang ¹ and Jianshan Chen ¹

¹College of Computer and Data Science, Fuzhou University, Fujian, China

²Department of Computing, Hongkong Polytechnic University, Hong Hom, Hongkong, China

³Fujian Key Laboratory of Network Computing and Intelligent Information Processing, Fuzhou University, Fujian, China

Correspondence should be addressed to Yingya Guo; guoyingya90@163.com

Received 11 August 2021; Accepted 24 November 2021; Published 11 December 2021

Academic Editor: Dimitrios Katsaros

Copyright © 2021 Yingya Guo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Internet traffic classification (TC) is a critical technique in network management and is widely applied in various applications. In traditional TC problems, the edge devices need to send the raw traffic data to the server for centralized processing, which not only generates a lot of communication overhead but also leads to the privacy leakage and information security issues. Federated learning (FL) is a new distributed machine learning paradigm that allows multiple clients to train a global model collaboratively without raw traffic data sharing. The TC in a FL framework preserves the user privacy and data security by keeping the raw traffic data local. However, because of the different user behaviours and user preferences, traffic data heterogeneity emerges. The existing FL solutions introduce bias in model training by averaging the local model parameters from all heterogeneous clients, which degrades the classification accuracy of the learnt global classification model. To improve the classification accuracy in heterogeneous data environment, this paper proposes a novel client selection algorithm, namely, WCL, in federated paradigm based on a combination of model weight divergence and local model training loss. Extensive experiments on the public traffic dataset QUIC and ISCX have proved that the WCL algorithm obtains, compared to CMFL, superior performance in improving model accuracy and convergence speed on low heterogeneous traffic data and high heterogeneous traffic data, respectively.

1. Introduction

Internet traffic classification, which classifies network traffic into different classes, plays a significant role in network management, such as network anomaly detection, quality of service (QoS), network monitoring, and traffic engineering (TE). In recent years, numerous TC methods have been proposed to classify the Internet traffic and the methods can be mainly divided into three categories: port-based classification methods [1], payload-based classification methods [2], and machine learning- (ML-) based [3] methods. In these traditional classification methods, all the raw traffic data have to be uploaded to the server for centralized processing, which raises peoples' concerns about data security and user privacy.

FL [4, 5] is a new proposed distributed ML paradigm that can address the data privacy and security issues in TC. In a FL paradigm, the raw traffic data is kept in local clients for training and the clients share the learnt classification model instead of raw traffic data, which greatly preserves user privacy and data security. However, due to the different user behaviours and preferences, heterogeneity of client data emerges. The heterogeneous traffic data introduces bias in model training and degrades global model accuracy in FL, for existing federated solutions mainly average the local models from selected clients to obtain the global model. Therefore, how to alleviate the bias of high heterogeneous data in global model training in FL has become a hot research topic. The existing works mainly adopt to weighted averaging the local model parameters [6] or selecting the

clients according to the same sign counts between the local models and global model [7], which can hardly eliminate the high heterogeneity clients in model aggregation.

Based on the abovementioned problems, in this paper, we propose a client selection method in FL based on a combination of model weight divergence and client training loss. For eliminating the high heterogeneous clients in model aggregation, the server selects the clients with smaller model weight divergence. For improving the convergence speed in FL, the server selects the clients with larger training losses. This algorithm can improve the traffic classification accuracy and convergence speed for model training in FL by selecting the appropriate clients to participate in. The main contributions of this paper are as follows:

- (i) First, we study the TC problem in a federated paradigm for preserving the data security and user privacy
- (ii) Second, we propose a new client selection algorithm WCL to optimize the accuracy and convergence speed of the FL model. Specifically, we combine the weight divergence between the client and the global model and the loss value of the client training process when selecting the clients in model aggregation
- (iii) Third, we conducted extensive experiments on public datasets QUIC and ISCX to demonstrate the superiority of the WCL algorithm. The experimental results demonstrate that WCL can improve the classification accuracy under different environment settings. Moreover, the convergence speed can also be improved

The rest of this paper is structured as follows. Section 2 is the related works on FL and TC. Section 3 presents the introduction of FL and the problem description. In Section 4, we provide the details about the proposed client selection algorithm WCL. In Section 5, we use real traffic datasets to evaluate the WCL algorithm under different environment settings. Finally, we make a conclusion in Section 6.

2. Related Work

In this part, we summarize the related works of FL and TC in recent years.

2.1. FL. There are two scenarios in FL. For the first scenario where all clients participate, Konečný et al. [8–18] proposed that the client data may be highly heterogeneous, which will affect the convergence of the local update SGD and may lead to a decrease in the accuracy of the FL model. In order to reduce the model accuracy on the heterogeneity of data, Sahu et al. have proposed the second scenario where only partial clients participate in the FL [19, 20]. In this paper, we study the Internet traffic classification in a FL scenario with partial client participation. However, how to choose the appropriate clients to improve the model accuracy and convergence speed poses a big challenge.

Luping et al. [7] proposed an orthogonal method CMFL to prevent the uploading of training data from clients with

lower correlation. The CMFL method uses the global update in the previous iteration to estimate the global update in the current iteration. This article explains that since the training of the model usually converges smoothly, the difference between the two sequential global models should be small. Therefore, the CMFL algorithm uses orthogonal calculations between the model training update data of the current client and the global model update in the previous iteration to determine the correlation between the client model update and the global model update. However, the relationship between orthogonal calculation and correlation mentioned in the paper does not seem to be convincing. However, the CMFL method has obvious effects in reducing communication overhead, but the improvement in model accuracy is extremely limited. Specifically, in the client selection algorithm, the power-of-choice framework improves the convergence speed of the model and the effect is obvious.

Cho et al. [21] proposed power-of-choice, a client selection framework with high communication and computational efficiency. The power-of-choice framework has also been confirmed, which can greatly improve the convergence speed of the model while reducing communication overhead. But this framework also has the same problem, that is, the improvement of model accuracy is very limited. Specifically, in the client selection algorithm, the power-of-choice framework improves the convergence speed of the model and the effect is obvious. Zhang et al. [6] mentioned the influence of the degree of nonindependent (non-IID) of client data on FL model training. The highly heterogeneous data distribution caused by non-IID data will bring bias in model training and may lead to a decrease in the accuracy of the FL model. Therefore, Zhang et al. [6] proposed a new FL method CSFedAvg. The CSFedAvg method uses the weight divergence to identify the non-IID degrees of clients, and CSFedAvg selects client update data with lower degree of non-IID according to the weight divergence to train the global model. This algorithm improves the model accuracy and convergence speed to a certain extent, but the improvement effect is limited. Therefore, how to choose a suitable client to participate in the training is a research hotspot that has received wide attention in the FL algorithm.

2.2. TC. TC has a wide range of applications in network management, including traffic security monitoring and service quality monitoring. In recent years, numerous TC methods have been proposed, which can be divided into three categories: port-based classification methods, load-based classification methods, and ML methods.

- (1) Port-based traffic classification method: the port-based classification method uses the port number in the TCP/UDP header to classify different types of network traffic. Obviously, the port-based classification is very simple and convenient. However, most new network applications use random port technology for data transmission, which brings difficulties to classification. And Auld et al. [2] proposed that this method is prone to interference, and the classification accuracy obtained is extremely unsatisfactory

- (2) Payload-based traffic classification method: the payload-based classification methods can be roughly divided into 3 steps: (a) checking the content of the data packet, (b) parsing the data packet, and (c) obtaining the characteristic fields from the data packet. Auld et al. [2] showed that the classification accuracy of this method is extremely high (greater than 99%). However, with the development of network technology, most applications adopt load encryption technology. Moreover, the extraction of feature fields usually requires a huge amount of overhead. Therefore, the effectiveness of this method is limited
- (3) ML-based traffic classification method: the classification method using ML requires the client to send local raw traffic data to the server for centralized training. In this way, a large amount of communication overhead is generated, and it also leads to the information security problem resulting from privacy leakage. Moreover, due to the traffic data heterogeneity, this may affect the accuracy of the classification model

In this paper, we propose a new FL algorithm WCL. The WCL algorithm proposes a new client selection scheme to select the clients with low heterogeneity in model aggregation, thereby effectively improving the training accuracy and convergence speed of the global model.

3. Preliminary

In this section, for the ease of understanding, we first briefly state the main steps of a FL framework, and give the general framework of FL in Figure 1. Then, we introduce the definition of Internet traffic classification problem in FL.

3.1. FL. In general, we can decompose the communication process of FL into 4 steps as shown in Figure 1:

- (1) Server first broadcasts the global model to the local clients
- (2) The local clients download the global model and train the traffic classification model with local raw traffic data using ML methods, such as SVM and deep learning (DL)
- (3) The local client uploads the trained model parameters to the federated server
- (4) Finally, the local models from the clients are aggregated in server, which commonly uses FedAvg algorithm [4]

After learning the communication process of FL, we now formalize the description of the model training process in FL as follows. Given a client set $C = \{C_1, C_2, \dots, C_N\}$ and a client local dataset $D = \{D_1, D_2, \dots, D_N\}$, here, N is the number of clients. In the client set C , C_k represents the k th client. In the dataset D , D_k represents the local data of the k -th client. For a traffic classification model, the goal is to minimize the model parameter x of the average classification loss $f(x)$:

$$\text{minimize } f(x) = \frac{1}{N} \sum_{k=1}^N f_k(x), \quad (1)$$

where $f_k(x)$ represents the loss value of client k when using its local data to train the model.

In the model training iteration process, we choose stochastic gradient descent (SGD) to optimize the model. Then, the update of the model $x_k(i)$ in the i th round of communication can be formulated as follows:

$$x_k(i) = x_k(i-1) - \eta_i \nabla f_k(x_k(i-1)), \quad (2)$$

where η_i denotes the learning rate in round i and $\nabla f_k(x_k(i-1))$ denote the gradient function.

Then, after obtaining the client model parameters in the current communication process, the server needs to integrate the client model parameters and obtain the global model $x(i)$ formulate as follows:

$$x(i) = \frac{1}{N} \sum_{k=1}^N x_k(i). \quad (3)$$

3.2. Traffic Classification Problem Definition in FL. The traffic classification problem is a multiclassification problem. In FL, given the distributed traffic dataset $D_i = \{X_i, Y_i\}$ in each client i , we need to find a function W to make the predicted value $W(X_i)$ of the function as close to the target value Y_i as possible. The definition of notations is summarized in Table 1.

4. Algorithm Description

In this section, we will introduce a new FL algorithm WCL, which proposes a new client selection scheme to improve the accuracy and convergence speed of the global classification model. Specifically, we first present the framework of the WCL algorithm. Then, we introduce our proposed algorithm WCL in details.

4.1. An Overview of the Proposed Algorithm WCL. In order to facilitate understanding, we first introduced the framework of WCL, which is shown in Figure 2. We assume that there is a federated server and N clients. Each client has its generated raw traffic data, and the client does not need to send its own private data to the server. The framework of FL preserve the user privacy and data security. In each round of WCL communication, the main process can be summarized as follows.

- (i) Step 1: the federated server uploads the global model and broadcasts it to candidate clients
- (ii) Step 2: after the client receives the server model, it uses its local data to train its own local model
- (iii) Step 3: after the client local model is trained, each client calculates the weight divergence between its own local model parameters and the global model

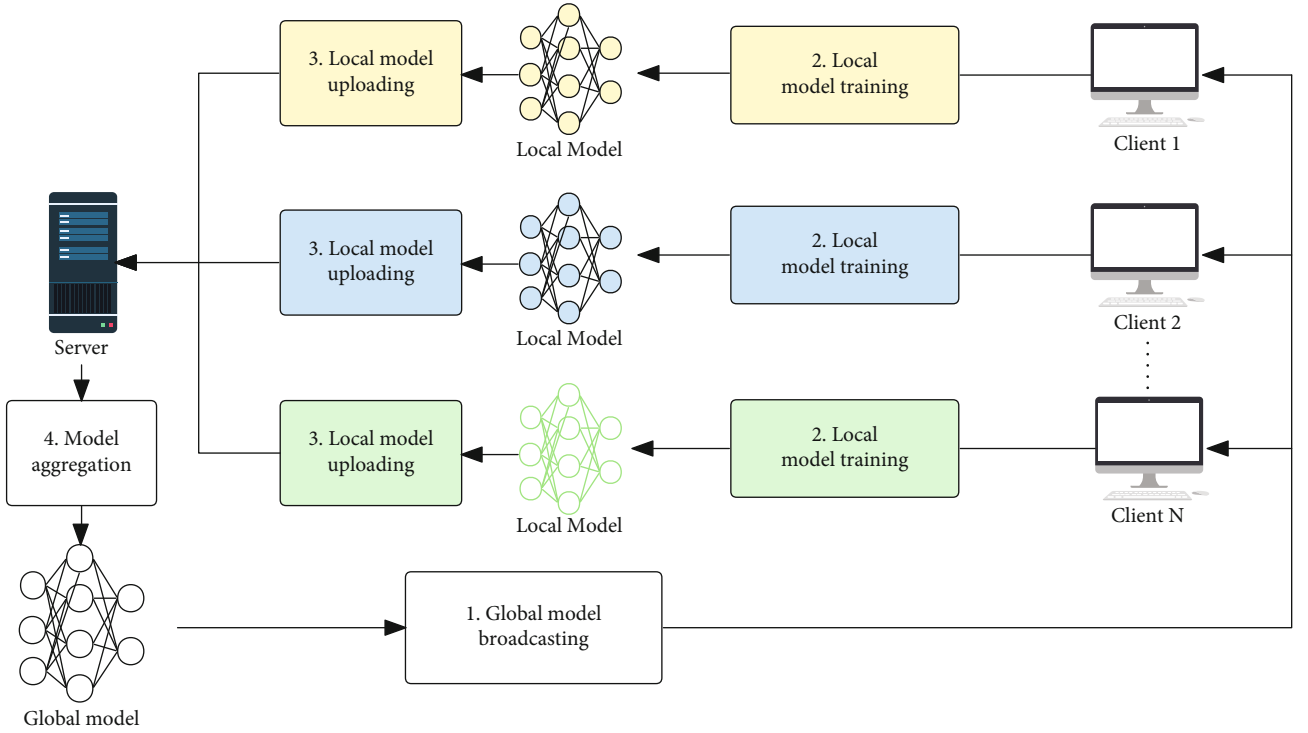


FIGURE 1: FL framework overview; the figure shows the basic framework structure and process of FL. It can be seen from the figure that FL generally has 4 steps: (1) The server broadcasts the global model. (2) The client downloads the model and uses local data to train the model. (3) The client uploads the model parameters. (4) The server aggregates the client model parameters.

TABLE 1: Definition of notations.

Symbol	Description
$C = \{C_1, C_2, \dots, C_N\}$	Candidate client set
$C_s = \{C_1, C_2, \dots, C_n\}$	Selected client set
$x(t)$	Global model obtained in round t
$x_k(t)$	The k th client model in round t
η_t	Client learning rate in round t
$\nabla f_k(x_k(t-1))$	Gradient function
$w_k(t)$	Weight divergence of the k th selected clients in round t
$l_k(t)$	Client training loss of the k th selected clients in round t
$p_k(t)$	Priority value of the k th selected clients in round t
m	The selected client ratio

parameters. Then, the clients send the weight divergence and the local model training loss to the server

- (iv) Step 4: the server considers the weight divergence and client loss returned by the client and obtains a priority value by considering two factors at the same time according to a certain weight ratio and selects a certain ratio of clients according to the priority value of each client

- (v) Step 5: the client selected to upload model parameters uploads its own local model parameters to the server

- (vi) Step 6: the server receives the model parameters from the selected clients, aggregates them, and obtains the global model

The above 6 steps form a communication round in the WCL algorithm. We need to iterate the above steps for multiple rounds until the target accuracy of the model is reached. It can be seen that compared to the traditional FL in which all clients participate, WCL algorithm adds two steps to each round of communication: (a) calculate weight bias and obtain client loss and (b) select the clients in FL.

4.2. Algorithm Details. We now introduce the WCL algorithm in details. The pseudocode of the WCL algorithm is shown in Algorithm 1. The input of the algorithm includes client set $C = \{C_1, C_2, \dots, C_N\}$, the weight coefficients of the weight divergence r_1 and the weight coefficients of client loss r_2 , the initial global model of the federated server: $x(0)$, selected clients ratio m , the number of candidate clients N , and learning rate η . The output of the WCL algorithm is an optimized global model x . Then, the main process of WCL is shown as follows.

The federated server first broadcasts the initial global model $x(0)$ to the candidate clients.

Then, the client receives the global model from the server and uses the local raw traffic data to train its local

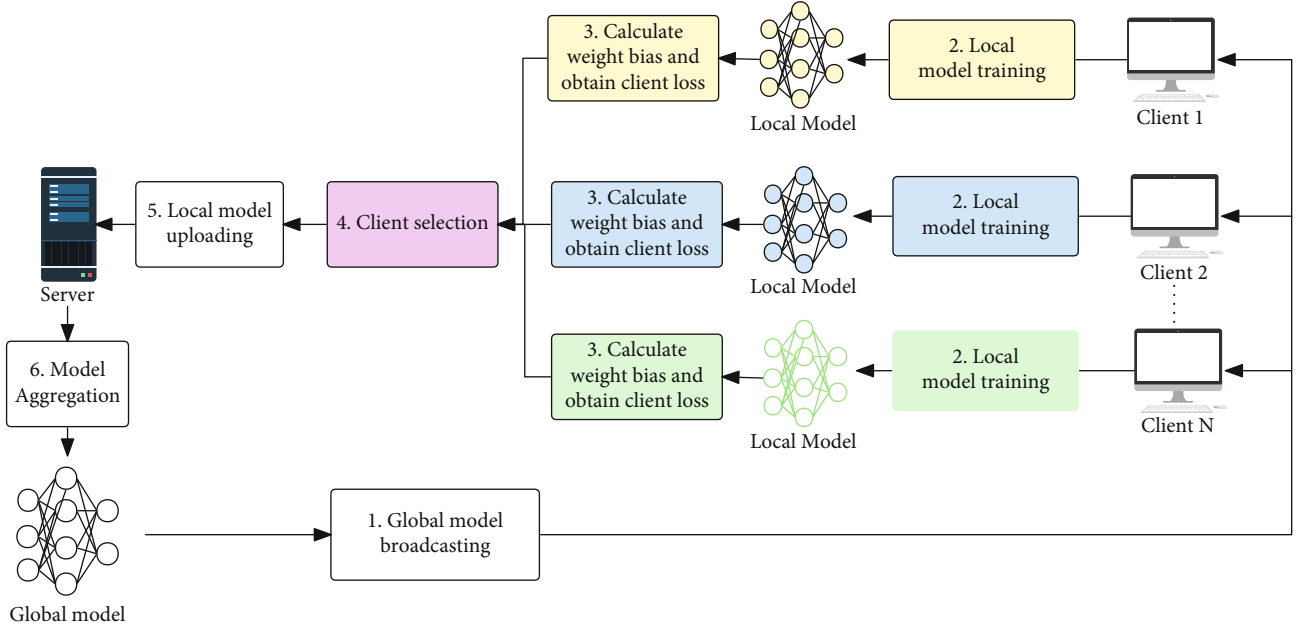


FIGURE 2: An overview of the proposed algorithm WCL. It can be seen from the figure that WCL has two more steps compared to FL in which all clients participate: (a) calculate weight divergence (bias) and obtain client loss and (b) client selection.

model. We set the client model for round t as $x_k(t)$. We can formulate the whole process as follows:

$$x_k(t) = x_k(t-1) - \eta_t \nabla f_k(x_k(t-1)), \quad (4)$$

where η_t denotes the client learning rate in round t and $\nabla f_k(x_k(t-1))$ denotes the gradient function. It can be clearly seen that $x_k(t-1)$ is consistent with $x(t-1)$.

After the client local model is trained, each client calculates the weight divergence $w_k(t)$ between its own local model parameters $x_k(t)$ and the global model parameters x_t . Since the optimization of the model is usually smooth, we believe that the difference between the model parameters of two adjacent rounds in sequence is small. Therefore, we use the global model of the previous round of communication $x(t-1)$ to calculate the weight divergence between the current round of the client model and the global model. Then, the client sends the weight divergence $w_k(t)$ and the client loss $l_k(t)$ to the server. Since the weight divergence and the client loss are rational numbers with small values, the communication overhead can be almost ignored compared to the model parameters. Then, we can formulate the weight divergence calculation as follows:

$$w_k(t) = \frac{\|x_k(t) - x(t-1)\|}{\|x(t-1)\|}. \quad (5)$$

The clients with less weight divergence and larger training loss are preferred in the client selection for alleviating the model bias incurred by the high heterogeneous clients, at the same time improving the converging speed.

The server considers both the model weight divergence $w_k(t)$ and client loss $l_k(t)$ uploaded by the clients and

obtains a priority value $p_k(t)$ by considering a linear combination of two factors at the same time. The value of $p_k(t)$ is computed as follows:

$$p_k(t) = r_1 w_k(t) - r_2 l_k(t), \quad (6)$$

where r_1, r_2 represent the weight coefficients of weight divergence value and client training loss value, respectively.

Once the priority value of the client is obtained, we can determine the set of selected clients C_s by selecting the clients with smaller values of $p_k(t)$.

$$C_s = \{C_1, C_2, \dots, C_n\}, \quad (7)$$

where n denotes the number of selected clients and m denotes the selected client ratio. The value of n is computed as follows:

$$n = m * N. \quad (8)$$

Then, the clients in C_s are selected to upload model parameters to the server for model aggregation.

Finally, the server receives the model parameters from the selected clients, aggregates the local models, and obtains the global model. In this paper, we leverage the most widely used method FedAvg to aggregate the client model. The process of averaging the uploaded local models is shown as follows.

$$x(t) = \frac{1}{n} \sum_{k=1}^n x_k(t). \quad (9)$$

The above process is repeated until reaching the

Input: Clients set C , the weight factor of the weight divergence and client loss r_1, r_2 , the initial global model of the federated server $x(0)$, selected clients ratio m during communication, number of candidate clients N , learning rate η .

Output: Optimized global model x

```

1: Initialize  $t = 0, x(0) \leftarrow x(t), C = \{C_1, C_2, \dots, C_N\}$ 
2: for each communication round  $t = 1, 2, \dots, T$  do
3:    $x(t-1) \leftarrow \text{federatedBroadcast}(x(t-1))$ 
4:   for each client  $C_k \in C$  do
5:      $x_k(t) \leftarrow \text{clientUpdate}(D_k, x(t-1))$ 
6:      $w_k(t) \leftarrow \text{getWeightBiasValues}(x(t-1), x_k(t))$ 
7:   end for
8:    $C_s \leftarrow \text{chooseClient}(w_{1..N}(t), m, l_{1..N}(t))$ 
9:   for  $\forall C \in C_s$  do
10:     $w_{1..n}(t) = \sum_{k=1}^n x_k(t)$ 
11:   end for
12:    $x(t) = 1/n \sum_{k=1}^n x_k(t) \leftarrow \text{getNewState}(C_s, w_{1..n})$ 
13: end for
14:
15: function CLIENTUPDATE $D_k, x(t-1)$ 
16:   for each local training epoch  $j = 1, 2, \dots, J$  do
17:      $x_k(t) = x_k(t-1) - \eta_t \nabla f_k(x_k(t-1))$ 
18:   end for
19:   return  $x_k(t)$ 
20: end function
21:
22: function CHOOSECLIENT $w_{1..N}(t), C_s, l_{1..N}(t)$ 
23:   for each client  $k = 1, 2, \dots, N$  do
24:      $p_k(t) = r_1 w_k(t) + r_2 l_k(t)$ 
25:   end for
26:    $C_{sIndex} \leftarrow \text{topCS}_r\text{OfClient}(p_{1..N}t)$ 
27:   return  $C_s$ 
28: end function

```

ALGORITHM 1: The proposed algorithm WCL in FL.

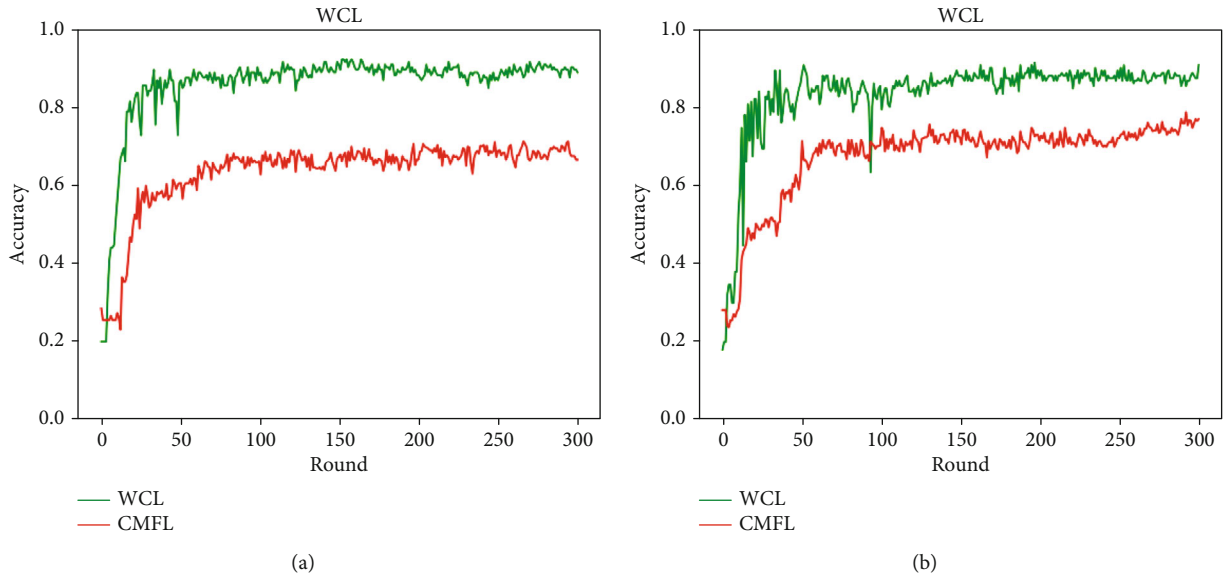


FIGURE 3: Accuracy curves of WCL under QUIC dataset with 30 clients and the selected client ratio set to 0.3 in two different data distribution situations: (a) high data heterogeneity and (b) low data heterogeneity.

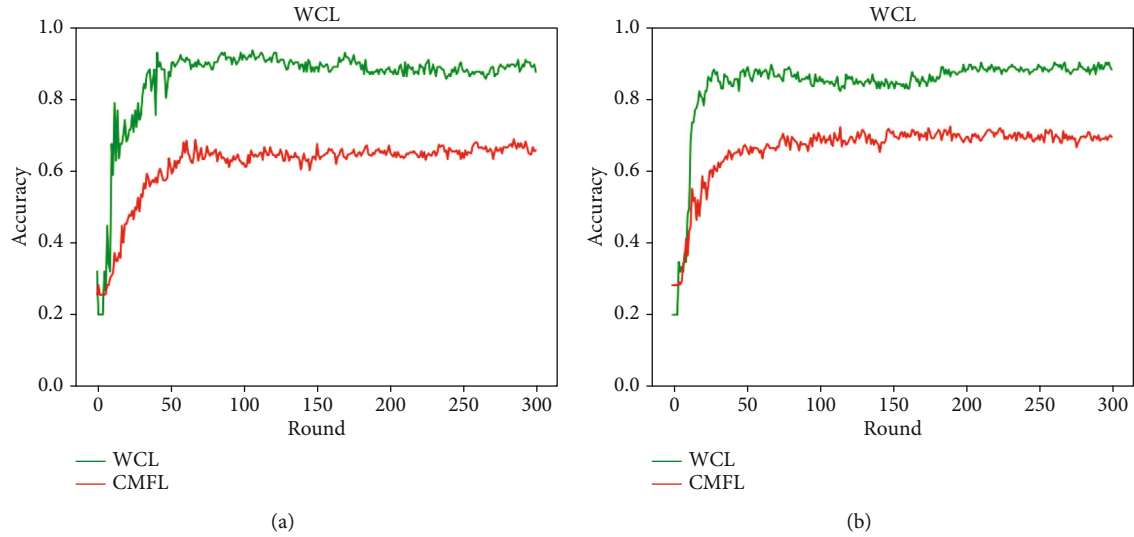


FIGURE 4: Accuracy curves of WCL under QUIC dataset with 40 clients and the selected client ratio set to 0.3 in two different data distribution situations: (a) high data heterogeneity and (b) low data heterogeneity.

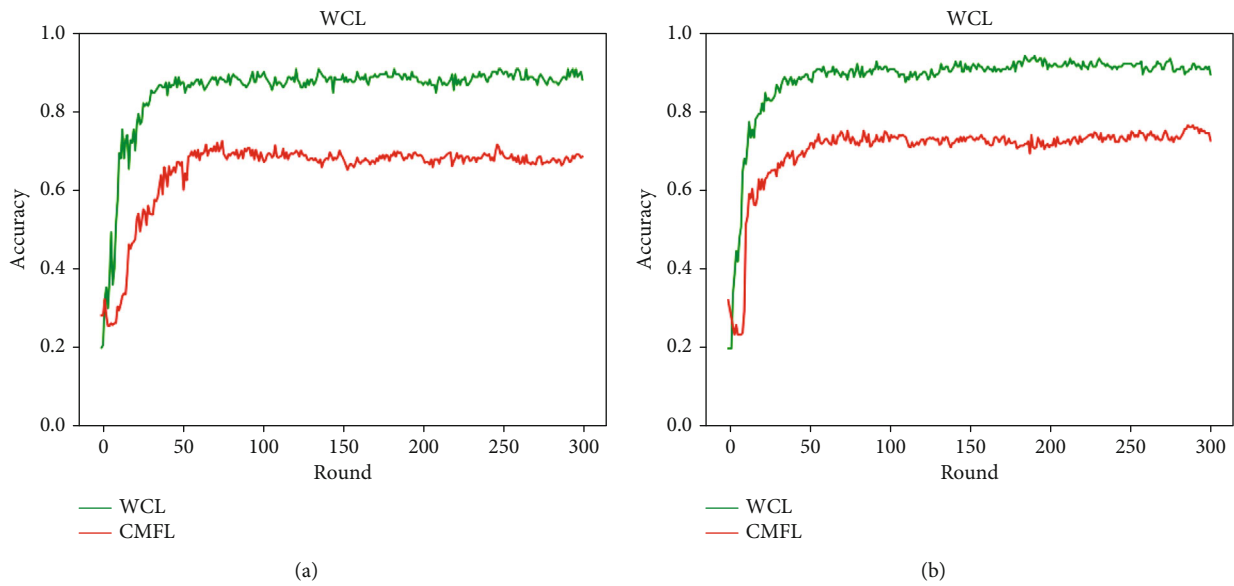


FIGURE 5: Accuracy curves of WCL under QUIC dataset with 50 clients and the selected client ratio set to 0.3 in two different data distribution situations: (a) high data heterogeneity and (b) low data heterogeneity.

maximum iteration times T . Finally, we can obtain a global classification model that improves the classification accuracy in heterogeneous environment.

5. Evaluation

In this part, we conducted extensive simulation experiments on public datasets to prove the superiority of the WCL algorithm. In the simulation experiment, we assume that all participating clients have the same computational capabilities. At the same time, we assume our system is a synchronized client-server system. All participating clients train the local model with the local traffic data and upload their weight

divergence and training loss to the server. The server must wait for all participating clients to accomplish the parameters uploading. We implemented the WCL algorithm using python and keras. During the evaluation process, the experiment was conducted on a personal computer equipped with a 2.8 GHz Intel Core i7-7700HQ and 8 GB of RAM.

The local learning models used by the WCL algorithm are the CNN models. The model consists of an input layer, convolutional layer, pooling layer, fully connected layer, and output layer. We set 3 layers of convolutional layer, the convolution kernel is a 3×3 convolution kernel, and the pooling parameter is set to 2. The three convolutional layers have 32, 64, and 128 convolution kernels,

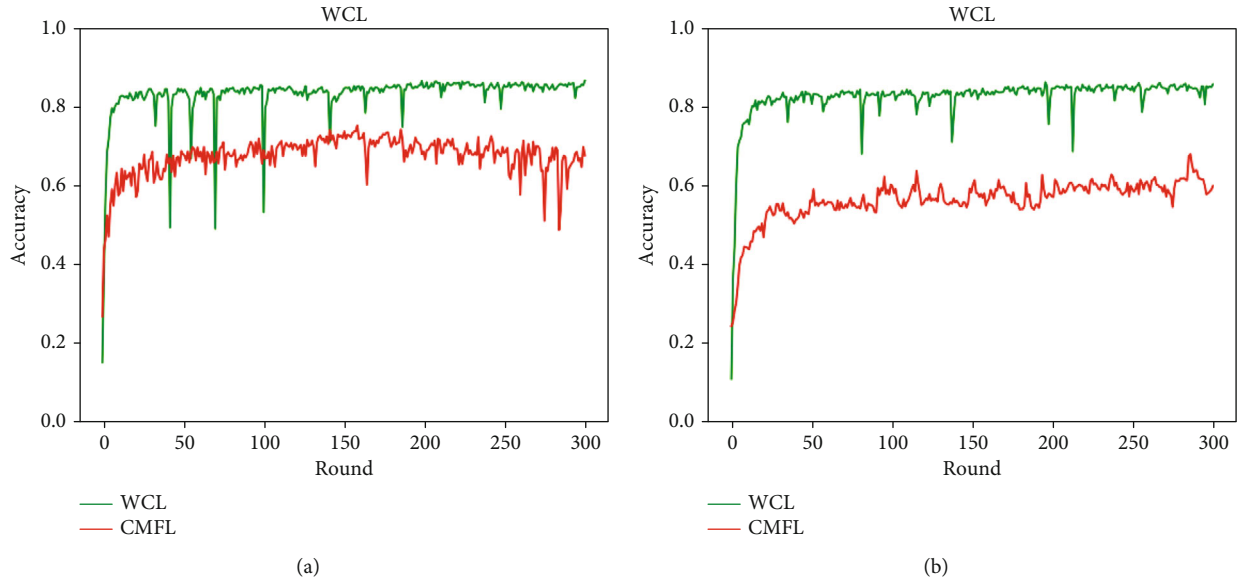


FIGURE 6: Accuracy curves of WCL under ISCX dataset with 30 clients and the selected clients ratio set to 0.3 in two different data distribution situations: (a) high data heterogeneity and (b) low data heterogeneity.

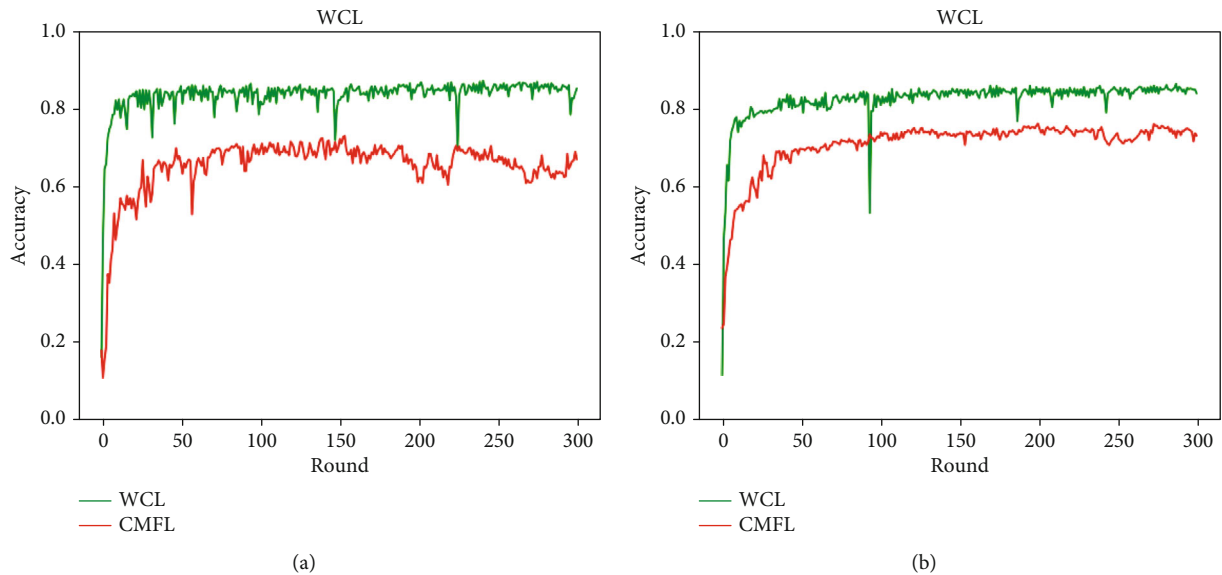


FIGURE 7: Accuracy curves of WCL under ISCX dataset with 40 clients and the selected client ratio set to 0.3 in two different data distribution situations: (a) high data heterogeneity and (b) low data heterogeneity.

respectively. And, in the convolutional layer and the fully connected layer, we choose ReLU as the activation function. In the output layer, we choose softmax as the activation function.

We used public traffic datasets QUIC and ISCX to validate the performance of WCL. The QUIC dataset includes 6589 traffic data samples, which can be classified into 5 categories. The ISCX dataset contains 60,000 traffic data samples, which can be divided into 14 categories. During the experiment, we set the number of communication between the client and the federation server to 300, the num-

ber of local iterations of the client to 10, and the ratio of the client to own the dataset to 0.8. For the number of clients, we have experiment with multiple sets of parameters. The number of clients are 30, 40, and 50, respectively. The selected clients ratios is set to 0.3. In addition, we set up two types of clients in the simulation experiment: (1) clients with smaller dataset distribution differences and (2) clients with larger dataset distributions. For the distribution of these two datasets, we also conducted a large number of simulation experiments to prove the superiority of the WCL algorithm. In order to demonstrate the improvement effect of

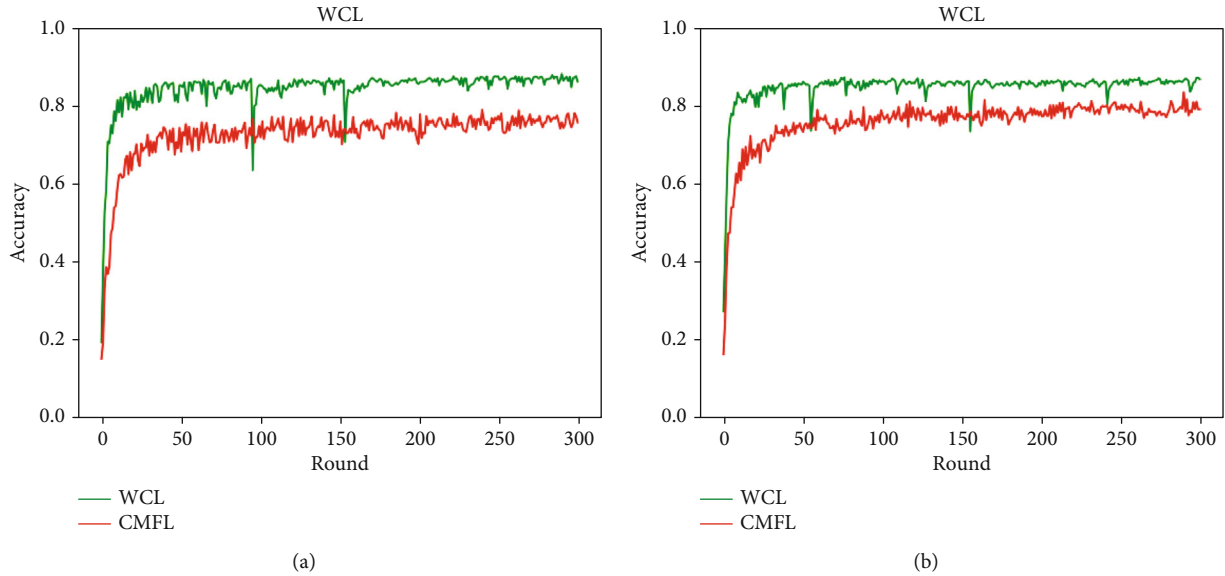


FIGURE 8: Accuracy curves of WCL under ISCX dataset with 50 clients and the selected client ratio set to 0.3 in two different data distribution situations: (a) high data heterogeneity and (b) low data heterogeneity.

TABLE 2: The improvement effect of the model accuracy of the WCL algorithm on the QUIC dataset.

Client number	Improvement on QUIC	
	High data heterogeneity	Low data heterogeneity
30	35.4%	23.9%
40	39.3%	26.7%
50	31.6%	25.5%

TABLE 3: The improvement effect of the model accuracy of the WCL algorithm on the ISCX dataset.

Client number	Improvement on ISCX	
	High data heterogeneity	Low data heterogeneity
30	23.1%	46.1%
40	27.9%	16.2%
50	16.3%	11.7%

the algorithm, we compare with the algorithm CMFL proposed in [7]. We plot the accuracy curves on the two datasets when the number of clients is set to 30, 40, and 50 in Figures 3–8.

As shown in Figures 3–8, we demonstrate the performance of the proposed WCL algorithm in improving the accuracy of the model and accelerating the convergence speed by comparing with the baseline. It can be seen from Figures 3–8 that whether it is on high heterogenous traffic data or low heterogenous traffic data, the WCL algorithm has a significant improvement in improving model accuracy compared with the CMFL algorithm. Similarly, in Figures 3–8, it can also be clearly observed that compared to the CMFL algorithm, the WCL algorithm has a significant improvement in model accuracy and convergence speed.

In Tables 2 and 3, we give a summary of the improvement ratios of the WCL algorithm on the two datasets. As shown in Table 2, the performance of WCL on the QUIC dataset is summarized. When the number of clients is 30, the accuracy improvement ratio of the WCL algorithm compared to the CMFL on the traffic data with different heterogeneities is 35.4% and 23.9%, respectively; when the number of clients is 40, the accuracy improvement ratio of the WCL algorithm compared to the CMFL on the traffic data with different heterogeneities is 39.3% and 26.7%, respectively; when the number of clients is 50, the accuracy improvement ratio of WCL compared to the CMFL on the traffic data with different heterogeneities is 31.6% and 25.5%, respectively. Table 3 summarizes the performance of WCL on the ISCX dataset. When the number of clients is set to 30, the improvement effect of WCL on the two distributions is 23.1% and 46.1%, respectively; when the number of clients is set to 40, the improvement effect of WCL is 27.9% and 16.2%, respectively; when the number of clients is set to 50, the WCL improvement effect is 16.3% and 11.7%, respectively.

In addition, we can also derive some conclusions related to the degree of data heterogeneity in Tables 2 and 3. The results in Tables 2 and 3 show that when the data has high heterogeneity, the improvement ratio of WCL compared to CMFL is higher than the improvement ratio when the data is in low heterogeneity. Combining the analysis of Figures 3–8 with Tables 2 and 3, it can be seen that when the data has high heterogeneity, the performance of CMFL is much worse than that has low heterogeneity data. In summary, WCL demonstrates its efficiency regardless of the high heterogeneous data or the low heterogeneous data. The WCL algorithm is not only suitable for low heterogeneous data but also for highly heterogeneous data. This also further reflects the superiority of the WCL algorithm.

6. Conclusion

This paper proposes a novel client selection algorithm WCL in FL to improve the training accuracy and convergence speed of the global classification model. Because the high heterogeneous client data will affect the accuracy and convergence speed of the federation model, in this paper, we propose a new client selection scheme based on weight divergence and client training loss in the WCL algorithm. In the selection process, the weight divergence reflects the degree of heterogeneity of client data. At the same time, considering the selection divergence of the client with higher local training loss makes the model converge faster. We demonstrate that a combination of weight divergence and local training loss of the clients can greatly improve the accuracy of the model and the speed of convergence when selecting the clients. In the evaluation, extensive experiments are conducted on the public traffic datasets QUIC and ISCX. The simulation experiment results show that the WCL algorithm greatly improves the training accuracy and convergence speed of the global classification model. Moreover, the WCL algorithm not only performs well in the environment with low heterogeneity data but also performs well in the environment with high heterogeneity data.

Data Availability

The traffic dataset QUIC is available at <https://drive.google.com/drive/folders/1Pvev0hJ82usPh6dWDlz7Lv8L6h3JpWhE>. The traffic dataset ISCX is available at <https://www.unb.ca/cic/datasets/vpn.html>.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work is partially supported by the National Key R&D Program of China under Grant No. 2020YFE0200500, National Natural Science Foundation of China under Grant No. 62002064, and Natural Science Foundation of Fujian Province under Grant 2020J0511.

References

- [1] A. Madhukar and C. Williamson, "A longitudinal study of P2P traffic classification," in *14th IEEE international symposium on modeling, analysis, and simulation*, pp. 179–188, Monterey, CA, USA, 2006.
- [2] T. Auld, A. W. Moore, and S. F. Gull, "Bayesian neural networks for internet traffic classification," *IEEE Transactions on Neural Networks*, vol. 18, no. 1, pp. 223–239, 2007.
- [3] J. Dean, G. S. Corrado, R. Monga et al., "Large scale distributed deep networks," in *Proceedings of the International Conference on Neural Information Processing Systems*, pp. 1223–1231, Mountain View, CA, 2012.
- [4] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Arcas, "Communication-efficient learning of deep networks from decentralized data," *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, vol. 54, pp. 1273–1282, 2017.
- [5] J. Konecny, B. McMahan, and D. Ramage, "Federated optimization: distributed optimization beyond the data center," 2015, <http://arxiv.org/abs/1511.03575>.
- [6] W. Zhang, X. Wang, P. Zhou, W. Wu, and X. Zhang, "Client selection for federated learning with non-IID data in mobile edge computing," *IEEE Access*, vol. 9, pp. 24462–24474, 2021.
- [7] W. Luping, W. Wei, and L. I. Bo, "CMFL: mitigating communication overhead for federated learning," in *2019 IEEE 39th international conference on distributed computing systems (ICDCS)*, pp. 954–964, Dallas, TX, USA, 2019.
- [8] J. Konečný, M. M. HB, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: strategies for improving communication efficiency," 2016, <http://arxiv.org/abs/1610.05492>.
- [9] S. Reddi, Z. Charles, M. Zaheer et al., "Adaptive federated optimization," 2020, <http://arxiv.org/abs/2003.00295>.
- [10] F. Haddadpour and M. Mahdavi, "On the convergence of local descent methods in federated learning," 2019, <http://arxiv.org/abs/1910.14425>.
- [11] A. Khaled, K. Mishchenko, and P. Richtárik, "Tighter theory for local SGD on identical and heterogeneous data," *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, vol. 108, pp. 4519–4529, 2020.
- [12] S. U. Stich and S. P. Karimireddy, "The error-feedback framework: better rates for SGD with delayed gradients and compressed communication," 2019, <http://arxiv.org/abs/1909.05350>.
- [13] B. Woodworth, K. K. Patel, S. U. Stich et al., "Is local SGD better than minibatch SGD?," 2020, <http://arxiv.org/abs/2002.07839>.
- [14] A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi, and S. U. Stich, "A unified theory of decentralized SGD with changing topology and local updates," p. 9, 2020, <http://arxiv.org/abs/2003.10422>.
- [15] Z. Huo, Q. Yang, G. Bin, L. Carin, and H. Huang, "Faster on-device training using new federated momentum algorithm," 2020, <http://arxiv.org/abs/2002.02090>.
- [16] X. Zhang, M. Hong, S. Dhople, W. Yin, and L. Yang, "FedPD: a federated learning framework with optimal rates and adaptivity to non-IID data," 2020, <http://arxiv.org/abs/2005.11418>.
- [17] R. Pathak and M. J. Wainwright, "FedSplit: an algorithmic framework for fast federated optimization," 2020, <http://arxiv.org/abs/2005.05238>.
- [18] G. Malinovsky, D. Kovalev, E. Gasanov, L. Condat, and P. Richtárik, "From local SGD to local fixed point methods for federated learning," 2020, <http://arxiv.org/abs/2004.01442>.
- [19] A. K. Sahu, T. Li, M. Sanjabi, M. Zaheer, A. Talwalkar, and V. Smith, "Federated optimization for heterogeneous networks," 2019, <https://arxiv.org/abs/1812.06127>.
- [20] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," 2020, <http://arxiv.org/abs/2007.07481>.
- [21] Y. J. Cho, J. Wang, and G. Joshi, "Client selection in federated learning: convergence analysis and power-of-choice selection strategies," 2020, <http://arxiv.org/abs/2010.01243>.