FedCPF: An Efficient-Communication Federated Learning Approach for Vehicular Edge Computing in 6G Communication Networks

Su Liu, Jiong Yu, Xiaoheng Deng, *Member, IEEE*, and Shaohua Wan, *Senior Member, IEEE*

***Abstract*— The sixth-generation network (6G) is expected to achieve a fully connected world, which makes full use of a large amount of sensitive data. Federated Learning (FL) is an emerging distributed computing paradigm. In Vehicular Edge Computing (VEC), FL is used to protect consumer data privacy. However, using FL in VEC will lead to expensive communication overheads, thereby occupying regular communication resources. In the traditional FL, the massive communication rounds before convergence lead to enormous communication costs. Furthermore, in each communication round, many clients upload large quantity model parameters to the parameter server in the uplink communication phase, which increases communication overheads. Moreover, a few straggler links and clients may prolong training time in each round, which will decrease the efficiency of FL and potentially increase the communication costs. In this work, we propose an efficient-communication approach, which consists of three parts, including "Customized", "Partial", and "Flexible", known as FedCPF. FedCPF provides a customized local training strategy for vehicular clients to achieve convergence quickly through a constraint item within fewer communication rounds. Moreover, considering the uplink congestion, we introduce a partial client participation rule to avoid numerous vehicles uploading their updates simultaneously. Besides, regarding the diverse finishing time points of federated training, we present a flexible aggregation policy for valid updates by constraining the upload time. Experimental results show that FedCPF outperforms the traditional FedAVG algorithm in terms of testing accuracy and communication optimization in various FL settings. Compared with the baseline, FedCPF achieves efficient communication with faster convergence speed and improves test accuracy by 6.31% on average. In addition, the average communication optimization rate is improved by 2.15 times.**

***Index Terms*— Federated learning, vehicular edge computing, 6G, efficient-communication, convergence speed, communication overhead.**

Manuscript received April 27, 2021; revised June 26, 2021; accepted July 15, 2021. This work was supported in part by the National Natural Science Foundation of China Project under Grant 61772553 and Grant 61862060 and in part by the Xinjiang Uygur Autonomous Region Graduate Research and Innovation Project under Grant XJ2021G063. The Associate Editor for this article was S. Mumtaz. *(Corresponding authors: Jiong Yu; Xiaoheng Deng.)*

Su Liu and Jiong Yu are with the School of Information Science and Engineering, Xinjiang University, Urumqi 830001, China (e-mail: liusu@stu.xju.edu.cn; yujiong@xju.edu.cn).

Xiaoheng Deng is with the School of Computer Science and Engineering, Central South University, Changsha 410083, China (e-mail: dxh@csu.edu.cn).

Shaohua Wan is with the School of Information and Safety Engineering, Zhongnan University of Economics and Law, Wuhan 430073, China (e-mail: shaohua.wan@ieee.org).

Digital Object Identifier 10.1109/TITS.2021.3099368

I. Introduction

THE tight integration of 6G and Vehicular Edge Computing (VEC) [1] spawns future vehicular networks [2], which have the potential to support autonomous driving and some vehicular applications [3], [4]. As an important emerging technology in 6G, Intelligent Transportation Systems (ITS) combines Artificial Intelligence (AI) [5] with VEC, further extending the potential value of vehicular-side data. With the rapid development of VEC technology [6], [7], vehicles and roadside units (RSUs) [8] at the network edge contribute much computation resources, storage capacity and rich data to handle AI tasks [9]. Recently, as the awareness of individual privacy protection has increased [1], the traditional data-driven centralized learning is not suitable for VEC scenarios. Proposed by Google [10], FL is a burgeoning distributed machine learning paradigm to avoid the privacy issues mentioned above. With the above advantage, FL has been applied broadly in VEC, such as traffic data collection [11], traffic flow prediction [12], [13] and urban traffic management [14]. Due to the continuous improvement of privacy protection awareness, vehicular users do not want to expose their driving habits, or information [15] to other users or upload it to the cloud server for analysis. Therefore, in the VEC scenario, FL is an appropriate technology to address this privacy issue [16]. Moreover, it use large-scale vehicular-side data to analyze user driving habits avoiding dangerous driving behavior. Massive vehicular data sets have profound impacts on the application of ITS [17], [18], which makes the ITS supported by 6G more reliable, efficient, and safer [19].

Despite potential benefits, a significant problem of the original FL is the issue of high communication overhead, which originates from the following three aspects. Firstly, the FL training process is very long, which means that there are many communication rounds before convergence. In each communication round, there are interactions between vehicles and the parameter server. Each vehicle receives the feedback information from the server and then updates its training result to the server. As the number of communication rounds increases, the overall communication overhead also increases dramatically. Existing research work shows that slower convergence speed increases communication rounds, proposing a trade-off between local computing and global communication [20]. Several current research works focus

on increasing convergence speed by adding local computing. As a result, communication rounds are decrease, then the communication cost declines [21], [22].

Secondly, the existing aggregation procedure of FL also affects the overall communication overhead. Due to many devices participating in the model aggregation phase, the communication cost increases with the number of devices imposing more pressure on communication channels. Therefore, several recent studies select a part of clients to participate in the aggregation phase [23], [24] for reducing the communication cost of each round. Besides, some studies [25], [26] consider data quality as conditions for selecting devices to join the aggregation to reduce the number of devices participating in the aggregation process.

Thirdly, the transmission validity between vehicles and the parameter server also affects the total communication costs. A small number of slower devices and unreliable wireless links, referred to as straggler links and devices, may drastically prolong training time and increase the communication costs of each round [27]. The study [28] considers that if the network connection is unstable, then part of the transmission content is invalid. To better meet the needs of the scenario, existing research work [29] proposes an adaptive aggregation strategy to reduce the effect of the stragglers by eliminating them. Once a client is selected to participate in the aggregation in the traditional FL, the aggregation will not end until all selected vehicles upload their parameters.

Therefore, the discussion mentioned above motivates us to investigate an efficient-communication method with a realistic architecture for vehicular networks to reduce the communication overheads with a guarantee of convergence speed. In this work, we propose a hierarchical FL approach to satisfy VEC, which is deployed at vehicular networks to alleviate privacy leakage, ensuring the services provided to passengers are private and safe. An efficient-communication FL algorithm, FedCPF, is integrated into this approach to tackle the expensive communication costs. To achieve a better trade-off between local computation and communication overheads, we allow the varied local training epochs for different clients, limiting their updated directions as close as possible to the global model. A customized local training strategy is employed to decrease communication rounds during the federated training phase. By adding a constraint item in the objective function, we optimize the local training epochs and increase the convergence speed. As the amount of data on each client is various, we should focus on extracting clients with a large amount of data to participate in the aggregation phase, summarized as a partial client participation rule for decreasing the communication overhead of each round. Additionally, we introduce a deadline for each client, limiting the time of transmitting the training results to the parameter server. Then the number of participants varies dynamically based on the number of clients who satisfy the aggregation phase requirements. Therefore, to reduce the communication costs, a flexible aggregation strategy is introduced to drop clients, which does not fulfill the upload time restriction.

In summary, the major contributions of our work are as follows.

- We introduce a customized local training strategy with a constraint item in the objective function on local clients to improve the convergence speed for reducing the total communication rounds.
- We propose a partial client participation rule which allows a few clients with massive local data to upload their training results simultaneously, decreasing communication costs in each round.
- We provide a flexible aggregation policy to drop the clients who exceed the time limitation to dynamically adjust the number of clients during the aggregation phase, thereby reducing each communication overheads.
- The experimental results illustrate that our approach effectively decreases the communication overheads in vehicular networks, compared with the traditional FL algorithm.

The rest of this paper is organized as follows: Section II discusses the related work. In section III, we propose a hierarchical FL network architecture. The FedCPF method is proposed to improve the expensive communication overheads in VEC during the learning phase in Section IV. In Section V, there are several experimental results under different experimental settings. Finally, Section VI comes to the conclusion.

II. Related Work

The expensive communication in distributed networks, especially in vehicular networks, remains the critical bottleneck of FL [30]. To improve communication efficiency, there are three main aspects of related works.

A. Reducing the Communication Rounds

The most commonly used algorithm for reducing the total number of communication rounds is Federated Averaging (FedAvg) [20]. This method enabled each client to communicate with the parameter server periodically. To reduce the communication rounds, each client performed several epochs before interacting with the server. Researchers set the local training epochs, denoted as E, without changing during the FL process. Moreover, the training epoch on different clients was the same E. The various devices updated towards diverse directions so that the convergence speed of the global model was slow. Another work [31] adopted a two-stream model with the maximum mean discrepancy constraint, which integrated more knowledge from the local model to the global one. This method increased the number of computations for local clients whereas reduced the overall communication rounds. The study [32] proposed a fast-convergent FL algorithm, called FOLB, which significantly reduced the number of communication rounds to reach a certain level of accuracy with an expected convergence speed. Although these methods decreased the global communication rounds, they did not consider the heterogeneity of the data of different clients. Consequently, a reasonable solution is to perform different local computations on various devices. So, we propose a customized local training strategy to constrain the divergence directions, which reduces the communication rounds.

B. Decreasing the Number of Upload Clients

Some studies focused on how to reduce the size of one communication round. On the one hand, model compression schemes such as sparsification [33], quantization [34] reduce the size of the transmitting messages in one communication round. For instance, several works provided practical strategies in federated settings, such as forcing the updating models to be sparse [35] and low-rank [36] to reduce the communication overheads from device to parameter server. Nevertheless, these researches did not consider the loss with the compression. On the other hand, another method for reducing transmitted messages was to decrease participation devices in the uplink communication phase. Wang *et al.* [37] provided feedback information in a communication-mitigated FL algorithm, which avoided irrelevant updates. Yang *et al.* [38] proposed an algorithm that the server chose only a subset of the devices in each communication round to decrease the number of upload clients. Ye *et al.* [25] proposed a selective aggregation method based on the heterogeneity of data in vehicular clients. Above all, the amount of data uploaded in one communication round decreases, but the client selection methods were not suitable for the demand of scenarios. So we propose a partial client participation rule, which allows a few clients to participate in the aggregation phase based on the data amount on the vehicles.

C. Improving the Dynamic of the Aggregation Mechanism

Considering the dynamic of devices, particularly in the federated network, multiple devices may be slow or completely inactive during the training procedure. Thus, the aggregation process should be flexible enough to adapt to the dynamical of the clients to improve the FL efficiency. Ruan *et al.* [39] proposed a flexible participation pattern for the devices. Specifically, it recommended removing inactive devices, which frequently had an unstable connection with the parameter server. Hence, this method decreased the number of devices to aggregate in one communication round. Zhuang *et al.* [40] put forward a FedPav strategy and a Cosine Distance Weight (CDW) method to dynamically decide whether the clients should join the current aggregation phase depending on the degree of changes between two successive updates. If the change was noticeable, then these devices need to join in the current aggregation phase. Bonawitz *et al.* [41] proposed a pace steering strategy, a flow control mechanism regulating the pattern of device connections. It was based on a simple mechanism that the parameter server received devices within an optimum time window to scale down the number of participants. Based on the above analysis, we propose a flexible aggregation policy in VEC, decreasing the communication costs in one communication round.

III. Federated Learning in VEC

In this section, we first summarize the typical FL procedure and then introduce a hierarchical FL approach among vehicular network, which applied to two various application scenarios.

A. Federated Learning Setup

In principle, the general setting in FL is that various clients have the same neural network model in the initial. These distributed devices train local models with their private data for a few epochs. For details, a complete communication round consists of the following phases:

(1) Downlink communication phase: The FL procedure starts with the downlink communication phase, which downloads the global model parameters $\boldsymbol{\omega}^t$ from the parameter server to K vehicles in the t-th round.

(2) Federated training phase: Each vehicle computes an updated model based on their collected data. In the traditional FL, it is worth noting that the federated training epochs remain the same across all clients. Even if different clients have different training data, varied clients still execute the same epochs in the federated training phase.

(3) Uplink communication phase: Many vehicles simultaneously transmit the updated model parameters to the parameter server through the uplink channels. This phase directly determines the size of communication overhead in each round. After multiple communication rounds, the number of participants has a crucial effect on total communication overheads in this phase. A downlink and an uplink communication phase constitute a complete communication round, ending with the uplink communication round.

(4) Aggregation phase: Each device transmits the latest model parameters to the parameter server. After the transmitted messages arrive at the parameter server, the aggregation phase starts. The server needs to wait for the transmitted data of all the participants sending to the parameter server, which may take a long time. Then the parameters in the same position of different models are added and averaged. The parameter server aggregates the global model and then distributes the updated output back to the devices. Consequently, inactive devices or stragglers will significantly slow down the aggregation phase.

In summary, each client repeats the above series of steps until the global model achieves convergence, then the process of FL stops.

B. System Framework

In VEC, there are many vehicular devices and RSUs, which produce a large quantity of data. Moreover, the distribution of these data sources is scattered. Consequently, it is more challenging to train a shared global model by multiple parties without sharing data. Considering the characteristics of FL, it satisfies the demands in VEC. As depicted in Fig. 2, we briefly review two kinds of vehicle status based on FL.

(a) In general, the vehicles collect massive data from the running process and then save them locally. For safety driving, no training task performs during the driving procedure. When the vehicle is in a parking state, it trains the local private data and collaboratively trains surrounding vehicles. For edge-based FL, the proximate RSU will act as the parameter server, while the vehicles within its communication range collaborate to train a shared global model. And then, vehicles upload the private training results to the RSU for aggregation [42] to complete a federated training round in VEC.

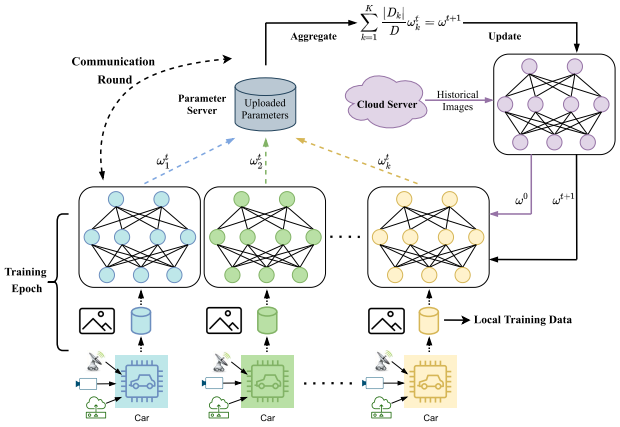


Fig. 1. Flowchart of original federated learning algorithm in VEC with traditional algorithm FedAvg.

(b) For vehicles in a long-driving status, they cannot acquire the parking conditions in the short term. The data on vehicles are transferred to RSUs through task offloading [43], while vehicles retain sufficient resources and storage space for driving [44]. The RSUs collect data from vehicles. Each RSU acts as a client and performs FL local training cooperatively with other RSUs. And then the RSUs upload the training outputs to the cloud server [45]. The cloud server is responsible for aggregation and then distribute the updated model. For cloud-based FL [46], the cloud server will act as the parameter server, while the participated RSUs provide massive data to form a complete FL round in VEC.

We propose a two-layers of network architecture, which contains a devices layer and a parameter server layer. This architecture is suitable for scenario (a) and scenario (b). In scenario (a), the devices layer is the vehicles, whereas the parameter server layer is the RSUs. In scenario (b), the devices layer is the RSUs with the data, whereas the parameter server layer is the cloud server. The federated training phase only performs at the devices layer without sharing the raw data by transmitting the private data to the server layer. The aggregation phase merely happens on the parameter server. From the above description, the FL architecture is proposed for vehicles in various states to provide a global model with good performance.

IV. Proposed Methods

In this section, a FL algorithm, named FedCPF, is introduced to count the total communication overheads of the training process. We give a review of the total communication costs of FL with a formulated upper bound. The FedCPF is proposed to reduce the total communication costs through the customized local training strategy, partial client participation rule, and flexible aggregation policy. After that, the convergence analysis for FedCPF is provided.

A. System Model

Suppose that there are K clients in total, where k is the index of the clients. Depicted as Fig. 1, each vehicle has a set of built-in radar, camera and sensors to capture data. The collected data is processed on the clients layer, which illustrated as Fig. 2. Each client k stores a local data set D_k, with its size represented by $|D_k|$, where $|\cdot|$ denotes the size of the set. Consequently, the whole data set is $\{D_1, D_2, \ldots, D_K\}$ across various clients, which has $|D| = \sum_{k=1}^{K} |D_k|$. There exists a local data set $\{x_k\}$ and the corresponding label set $\{y_k\}$ in D_k. In a typical learning problem, for a training data sample i, which belongs to the D_k, consists of two parts. One is a vector x_i that is the input of the local neural network, such as the pixels of an image. The other is a scalar y_i that is the desired output. The object is to find the model parameter $\boldsymbol{\omega}$ that characterizes the output y_i with the loss function $f_i(\boldsymbol{\omega})$. For client k, the loss function on the data set is defined as:

$$\min_{\boldsymbol{\omega}} F_k(\boldsymbol{\omega}) = \frac{1}{|D_k|} \sum_{i \in D_k} f_i(\boldsymbol{\omega}). \tag{1}$$

In a FL problem, the training process relies on the distributed stochastic gradient descent (DSGD) with various data sets. Moreover, FL aims to the above issue but in a distributed manner. At the beginning of the FL procedure, we initialize the local models with global parameters $\boldsymbol{\omega}^0$ through the historical data stored at the parameter server. $\boldsymbol{\omega}_k^t$ is the model parameters transferred between the devices layer and the parameter server layer, which denotes as the model parameters of k-th client in the t-th communication round in FL. At the t-th communication round, the objective function is to minimize the training error as depicted (2):

$$\min_{\boldsymbol{\omega}_k^t} F_k^t(\boldsymbol{\omega}_k) = \frac{1}{|D_k|} \sum_{i \in D_k} f_i\left(\boldsymbol{\omega}_k^t\right). \tag{2}$$

And we use $\boldsymbol{\omega}_k^t$ to find the minimizer of the loss function. As for the similar process of each communication round in FL, the number of communication rounds t is removed for simplicity. Furthermore, for each FL training round, we simply remove the index of t as the (3) is true:

$$\min_{\boldsymbol{\omega}_k} F_k(\boldsymbol{\omega}_k) = \frac{1}{|D_k|} \sum_{i \in D_k} f_i(\boldsymbol{\omega}_k). \tag{3}$$

For convenience, the objective function in FL can be reformulated as $F(\boldsymbol{\omega})$, $F_k(\boldsymbol{\omega}_k)$ denotes the loss function of k-th client. Then the federated learning tasks attempt to optimize the trainable parameter $\boldsymbol{\omega}$ by minimizing the $F(\boldsymbol{\omega})$. $F(\boldsymbol{\omega})$ denotes the averaged global model parameters after the aggregation phase. So, the loss function in equation (3) can be generalized as following equation (4):

$$\min_{\boldsymbol{\omega}} F(\boldsymbol{\omega}) = \sum_{k=1}^{K} \frac{|D_k|}{|D|} F_k(\boldsymbol{\omega}_k). \tag{4}$$

Moreover, optimizing the loss function $F(\boldsymbol{\omega})$ in FL is equivalent to minimizing the weighted average of local loss function $F_k(\boldsymbol{\omega}_k)$. Each client performs the training task locally, and shared the own local parameters.

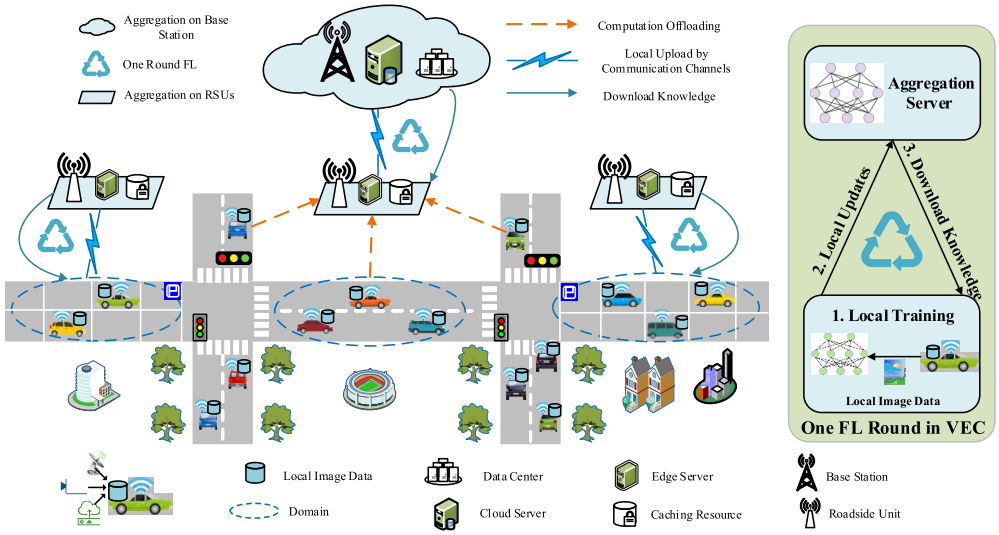


Fig. 2. Architecture in VEC with Federated Learning.

B. Total Communication Costs of FL

This section reviews the overall communication costs in the FL process that represents the critical metric of communication efficiency and formulates an upper bound to communication costs.

The FL process usually consists of two communication phases, the downlink communication phase, and the uplink communication phase. Let K be the total number of the devices, and $n(t)$ $(n(t) \leq K)$ be the number of devices participate in the t-th communication round. The transmitted data from these $n(t)$ clients should arrive at the parameter server in an effective time. T represents the total communication rounds in the FL process. Considering a homogeneous environment, each local device has the same model, which means the number of the model parameter is identical, denoted as ω^*. W represents the total communication costs in FL, and it is calculated as equation (5):

$$\begin{aligned} \mathrm{W} &= \sum_{t=1}^{T} \left\{ (n(t) \cdot \omega^*) + K \cdot \omega^* \right\} \\ &= \sum_{t=1}^{T} \{ n(t) + K \} \cdot \omega^*. \end{aligned} \tag{5}$$

We found that an upper bound of communication overhead of the standard FL process can be deduced. There are K clients in a loose situation to train the shared model collaboratively, and we assume that K clients all participate in the training phase and aggregation phase in every communication rounds. Under the ideal conditions, each device needs to upload the total parameters of the local model to the parameter server. After the aggregation phase, the updated parameters will be distributed to all clients. The transmission overhead is denoted as $\tilde{\mathrm{W}}$ which has an upper bound. To summarize, we represent the communication rounds as T and the complete transmission rounds as $2T$. Consequently, the upper bound of communication overheads is as equation (6):

$$\tilde{\mathrm{W}} = 2T(K \cdot \omega^*). \tag{6}$$

In equation (5), the total number of devices K is larger than $n(t)$. So, this is the upper bound of communication costs that we will never achieve in a general situation.

According to the above upper bound in formula (6), we optimize the communication overhead from the following points. Firstly, considering the long training process, we shorten the learning procedure to increase the convergence speed and decrease T. Secondly, considering the asymmetry between the downlink and the uplink communication channels, we reduce the amount of the participation devices during the FL process, choosing $n(t)$ in a reasonable way.

C. A Customized Local Training Strategy

Gradient diversity quantifies the degree to which individual gradients of the loss functions vary from each other clients in distributed computing. We note that this phenomenon happens in the FL process. We use T to denote the total number of communication rounds. Each client k has its local model parameter $\boldsymbol{\omega}_k^t$, where $t = 0, 1, 2, \ldots$ represents the communication round index. At the beginning of FL, all the clients have the same model parameters and identical model architectures. According to the previous work, the local parameters $\boldsymbol{\omega}_k^t$ at each client k will change after a global aggregation. Meanwhile, the gradients at different clients are updated towards varied directions due to the various distributions of the local data sets. Consequently, there is a weight divergence phenomenon between local models and the global model. As a convenience, we use $\boldsymbol{\omega}_k^t$ to denote the model parameter set of client k at t-th FL round. If aggregation is performed at t-th communication round, then generally $\boldsymbol{\omega}_k^{t+1} \neq \boldsymbol{\omega}_k^t$. And then we set $\boldsymbol{\omega}_k^{t+1} = \boldsymbol{\omega}^t$, where $\boldsymbol{\omega}^t$ is a weighted average of $\boldsymbol{\omega}_k^t$. As shown in Fig. 3, different clients start from the same initial $\omega_{(G)}^{Ini}$ in the FL, which is the same as

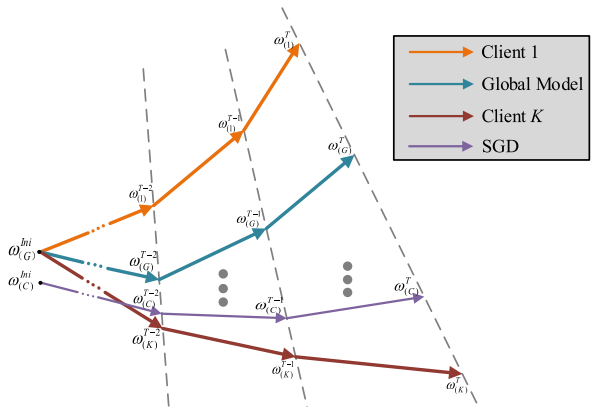


Fig. 3. Illustration of the weight divergence for FL.

the traditional centralized settings. The orange curve represents one of the clients participating in FL, and the red curve represents another client k participating in FL. As the number of communication rounds increases, the clients use different data sets to update the model parameters in different directions. In each communication round, after aggregation by the parameter server, different clients' parameters are added and then averaged to obtain a global model, which is represented by a blue curve in Fig. 3. The ideal situation is that the global model obtained by FL can be closer to the performance obtained by traditional centralized training as the purple curve in Fig. 3. But after T communication rounds, the weight divergence is significant. We suppose that there are C labels in the label space. The fundamental reason of the weight divergence is $\sum_{i=1}^{C} \left\| p^{(k)}(y=i) - p(y=i) \right\|$. For convenience, we use $\|\cdot\|$ to denote the $\mathcal{L}^2$ norm in this work. This situation is termed as Earth Mover's Distance (EMD) [47], which is between the data distribution on clients k and the whole data distribution. This distance measurement is defined as $\left\| p^{(k)}(y=i) - p(y=i) \right\|$, which is affected by the learning rate η, the number of training epochs, and the gradient. In the FL setting, the leading causes of weight divergence mainly derive from two aspects. (1). The weight divergence accumulates after the T-1 communication rounds. (2). Weight divergence is induced by the probability distance for the data distribution on client k compared with the actual distribution for the whole data set. Based on the above discussion, different clients have different resource conditions, for example, network connections, computing hardware, and data sets. If all local models perform the same epochs E on their local data sets, there will be a significant divergence of the local gradients. As a result, we need to set an appropriate local training strategy for each client with a different training amount. In FedCPF, we generalize a customized local training strategy by allowing customized training locally across clients.

Specifically, we introduce a constraint item to narrow down the gradients diversity among local functions effectively. Instead of only minimizing the local loss function $F_k(\cdot)$, client k minimize the following objective equation (7) approximately with its local loss function:

$$\min_{\boldsymbol{\omega}} g_k\left(\boldsymbol{\omega}_k; \boldsymbol{\omega}^t\right) = F_k\left(\boldsymbol{\omega}_k\right) + \frac{\varepsilon}{2}\left\|\boldsymbol{\omega}_k - \boldsymbol{\omega}^t\right\|^2. \tag{7}$$

Definition 1 (θ_k^t Customized Local Training): For the objective equation (7), we propose a variable θ to denote the varied local training epochs, and the θ_k^t denotes running θ epochs on client k at the t-th federated training. We suppose $\hat{\omega}_k$ is an intermediate solution of equation (6) if after θ_k^t local training ($\theta_k^t \leq E$), $\left\|\nabla g_k\left(\hat{\omega}_k; \omega^t\right)\right\| \leq \left\|\nabla g\left(\omega_k^t; \omega^t\right)\right\|$, where $\nabla g\left(\omega_k; \omega^0\right) = \nabla F_k\left(\omega_k\right) + \varepsilon\left(\omega_k - \omega^0\right)$. Note that a less local computation corresponds to higher accuracy.

This constraint item $\frac{\varepsilon}{2}\left\|\boldsymbol{\omega}_k - \boldsymbol{\omega}^t\right\|^2$ is beneficial from the two folds. On the one hand, it alleviates the statistical heterogeneity by limiting the number of local updates as possible as closer to the global model without manually setting the local training epochs. On the other hand, it allows for safely incorporating variable amounts of local training resulting from systems heterogeneity.

Too many local training epochs in one communication round may cause a slow convergence speed of the global model due to the potential heterogeneous of the private data. However, as mentioned above, the customized local training strategy allows for inconsistent local training epochs performed across clients, which effectively alleviates the negative impacts of gradient divergence.

D. Partial Client Participation Rule

In the VEC scenario, there are often many clients communicating with a parameter server. Two aspects will affect the FL efficiency. Firstly, considering the limitation of uplink communication channels, the clients usually have limited upload bandwidth. We do not consider the downlink communication time, when compared with the uplink one, is negligible. The reason is that the downlink has a larger bandwidth than the uplink. Hence, only a small group of devices simultaneously upload their training outputs to the parameter server. Due to the limitation, the training outputs sent from the devices will be pipelined at the parameter server, which results in dramatically slow training. Secondly, some clients only have a small amount of data, which will become noise in the aggregation phase. In that case, it will cause the global model to be more biased towards individuals rather than the common development direction of most clients. Therefore, we propose a partial client participation rule to decrease the number of clients in the uplink communication phase to decrease the $n(t)$.

At each global communication round, a subset of the clients, denoted as $\{S_t\}$, are selected. Local models on the clients are used to optimize the local objective functions with equation (4) on every selected client. Notably, in practical applications, not all the training results on the clients play a role in each round of training. In the uplink communication phase, clients upload local model parameters, also transmit the size of the local data set to the parameter server simultaneously. Based on the uploaded information, the parameter server calculates the probability of each client. The probability values represent the probability of clients being selected. We use p_k to represent the probability of the client k being selected in each round. The client with large amounts of local data tends to have a bigger p_k. The subset of selected clients in each round is different, making the global model contain as much

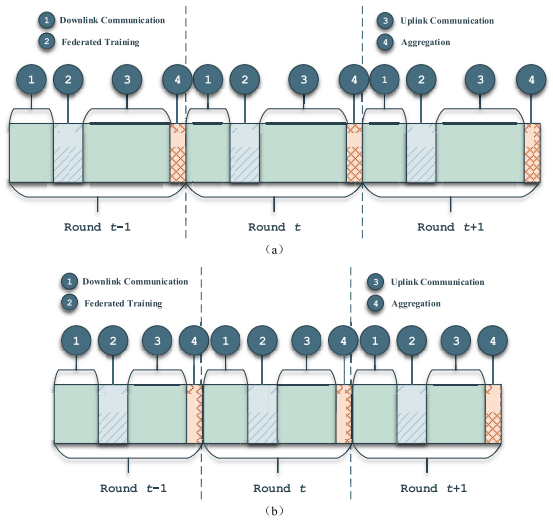


Fig. 4. A schematic diagram of a federated learning communication round.

information about local clients as possible. So, equation (4) and equation (7) are changed as equation (8):

$$\min_{\boldsymbol{\omega}} F(\boldsymbol{\omega}) = \sum_{k=1}^{|S_t|} p_k g_k\left(\boldsymbol{\omega}_k; \boldsymbol{\omega}^t\right), \sum_{k=1}^{|S_t|} p_k = 1 \tag{8}$$

The partial client participation rule contains more information about client data sets, avoiding developing the global model in a personalized direction. The number of clients participating in each round is reduced, so the communication costs in each round are significantly decreased.

E. A Flexible Aggregation Policy

Considering the local training process on clients is highly dynamic, we propose the flexible aggregation policy to keep in synchronicity. In the local training phase, the training completion time of the clients should be limited, which ensures that training outputs are transmitted to the parameter server in time and then enter the aggregation stage.

As in Fig. 4(a), we find that the most time-consuming phase is the third part, the uplink communication. At this phase, the parameter server needs to wait for all the clients to upload local training results and aggregate them. Therefore, we propose a flexible aggregation policy to limit the time of the uplink communication phase. Denoting t represents the t-th global communication round, the t-th uplink communication phase begins at the start time r_s^t, the t-th uplink communication phase finishes at the end time r_e^t. r_k represents the k-th client finishes the local training time and then transmits the parameters to the parameter server layer. To ensure that the parameter server receives the content from the client k normally, the transmission time should be restricted as equation (9):

$$r_s^t \leq r_k \leq r_e^t. \tag{9}$$

The above restriction makes the system more flexible, considering the clients can be dropped due to various reasons, such as low battery power, system failure, and accelerating the overall training process. As shown in Fig. 4(b), based on this flexible aggregation policy, the time of the uplink communication phase is shortened. The time of this FL round is also shortened overall, so the communication efficiency of the FL will be improved.

The running time of a complete FL round is represented as T_g. The communication time in one FL round represents by T_{co}. Denoting the time of one local training epoch as T_{cp}, the number of local training epochs is denoted as θ_k^t on the k-th client, then the computation time in one FL round is $\theta_k^t \cdot T_{cp}$. The running time of a FL round is defined as:

$$T_g = T_{co} + \theta_k^t \cdot T_{cp}. \tag{10}$$

We shorten the time of the uplink communication phase T_{co} by the flexible aggregation policy. Based on the flexible aggregation policy, the number of clients participating in each round of aggregation has shown a declining trend. The communication overhead of each round is reduced, which leads to the global communication overhead drops off. We summarize the whole procedure of FedCPF as Algorithm 1.

Algorithm 1 FedCPF (Proposed Approach)

Input: The K clients are indexed by k, toal communication rounds of FL T, the coefficient of the constraint item ε, the total number of the clients K, private data set on client k is $D_k, k = 1, \ldots, K$, $n(t)$ denotes the number of chosen clients at the t-th communication round, and constitutes a subset S_t, the number of local training epochs is θ_k^t, the initialized parameters of the model ω^0.

Output: The updated global model parameters ω_k^{t+1}

Initialize all the local models parameters with ω^0
for $t \leftarrow 0$ to $T-1$ **do**
  Downlink Communication:
  Server picks a subset S_t of K clients with the probability p_k.
  Server sends ω^t to clients in the subset S_t.
  Federated Training:
  for each client $k \in S_t$ **in parallel do**
    Each client computes the θ_k^t local epochs with equation (8) to find a minimum loss function through the constraint item.
  end for
  Uplink Communication:
  Each client in the subset S_t sends the ω_k^{t+1} back to the server before the defined time considering the flexible aggregation policy.
  Aggregation:
  Server average the ω^{t+1} with the probability p_k.
  Server distribute the ω^{t+1} to all the clients.
end for

F. Convergence Analysis

In this section, we provide the theoretical results on the guarantees of the FedCPF algorithm. Essentially, FedCPF is a stochastic algorithm. In each communication round, only a fraction of clients are chosen to perform the local training,

and the updates on each client may be imprecise. The degree of diversity among the varied local loss functions needs to be quantified in IID ideal federated networks to analyze the convergence in practice. We use the V-local diversity to measure the diversity among the clients in VEC, which is sufficient to prove convergence in FL. And the definition of this metric is as follows:

Definition 2 (V-Local Diversity): The local loss functions F_k are V-locally diversity at ω if the equation (11) is satisfied.

$$\mathbb{E}_k\left[\left\|\nabla F_k(\omega_k)\right\|^2\right] \leq V^2\left\|\nabla F_k(\omega_k)\right\|^2. \tag{11}$$

Here, we define the $\|\nabla f(\omega)\|^2 \neq 0$.

$$V(\omega) = \sqrt{\frac{E_k\left[\left\|\nabla F_k(\omega_k)\right\|^2\right]}{\|\nabla f(\omega)\|^2}} \neq 0. \tag{12}$$

And $\mathbb{E}_k[\cdot]$ represents the exception over clients with masses data as in equation (3). As a sanity check, equation (11) and equation (12) can be define as the following equation (13):

$$\mathbb{E}_k\left[\left\|\nabla F_k(\omega_k)\right\|^2\right] \leq V^2\left\|\nabla F_k(\omega_k)\right\|^2, \quad \begin{cases} V = 1, \text{if data is IID} \\ V > 1, \text{if data is Non-IID.} \end{cases} \tag{13}$$

We propose the formal diversity assumption based on Definition 2, which is used in the convergence analysis. This merely requires that the diversity defined in Definition 2 is bounded. Based on the above discussion, the convergence rate is a function of the statistical heterogeneity and the client diversity in the distributed network.

Assumption 1 (Bounded Diversity): For some $\rho > 0$, there exists a V_ρ such that for all $\omega \in S_\rho^c = \left\{\omega \middle| \|\nabla f(\omega)\|^2 > \rho\right\}$, there is $V(\omega) \leq V_\rho$.

For the practical FL scenarios, like VEC, the data distribution is Non-IID. Data on any client will not represent the entire data set distribution. Accordingly, it is reasonable to suppose that the diversity among local loss functions remains bounded. Furthermore, the experimental results also demonstrate the above assumption in Section V. To reduce the communication overheads, and only $n(t)$ clients are selected to join in each round. Using the bounded diversity assumption, we analyze the convergence of the FedCPF algorithm in Appendix A for detailed proof.

V. Experiments

This section presents the experimental results of the proposed FedCPF approach, which reduces the communication overheads by employing three modules in Section IV. In the first subsection, the details of the experimental setup are provided. Then, we demonstrate the improved accuracy and loss by allowing inexact local solutions under the varied statistical heterogeneity in the second subsection. In the third subsection, we show the effectiveness of FedCPF with various system heterogeneity. Finally, we show the total communication costs on varied data sets and the advantages of FedCPF over FedAvg.

TABLE I
Statistical Details of Five Used Federated Data Sets

data sets	Devices	Samples	Samples/devices	
			mean	stdev
Synthetic	100	12,697	127	73
MNIST	1,000	69,035	69	106
FEMNIST	200	18,345	92	159

A. Experimental Data Sets

We evaluate FedCPF on various models and data sets, including real-world FL data sets. To better simulate statistical heterogeneity and discuss the effect on convergence, we also assess a series of Synthetic data sets [21]. Considering the statistical heterogeneity, we assign a different number of local work to various devices for mimicking statistical heterogeneity in the Synthetic data sets.

The full details of data sets and models used in the experiments are proposed in this section, including a diverse group of non-synthetic data sets, and some proposed in LEAF [48], a benchmark for FL. Details of real data sets, statistics are summarized in Table I.

- Synthetic: To better generate synthetic data, we follow a similar setup as that in [49], which carrying out heterogeneity among diverse clients. For client k, we generate the samples (x_k, y_k) with the model $y = arg\max(\text{softmax}(\boldsymbol{\omega}x + b))$, $x \in \mathbb{R}^{60}$, $\boldsymbol{\omega} \in \mathbb{R}^{10\times 60}$, $b \in \mathbb{R}^{10}$. And then we pursue the global $\boldsymbol{\omega}$ and b. Samples (x_k, y_k) and local models on each client k satisfies $\omega_k \sim \mathcal{N}(u_k, 1)$, $b_k \sim \mathcal{N}(u_k, 1)$, $u_k \sim \mathcal{N}(0, \alpha)$; $x_k \sim \mathcal{N}\left(v_k, \sum\right)$, where the covariance matrix $\sum$ is diagonal with $\sum_{j,j} = j^{-1.2}$. Every elements in mean vector v_k is drawn from $\mathcal{N}(B_k, 1)$, $B_k \sim \mathcal{N}(0, \beta)$. Furthermore, α denotes how much local models differ from each other, and β represents how much the local data at each client differs from that of other clients. We construct different degrees of heterogeneity through three various value schemes of Synthetic_α_β. The IID data set is generated by setting the identical $\boldsymbol{\omega}$ and b on all clients and setting x_k to subject to the same distribution. We use the Synthetic data set to present different degrees of data heterogeneity and observe the experiment performance of various degrees of FL heterogeneity settings.
- MNIST: We perform an image classification task of handwritten digits 0-9 in MNIST with a multi-class logistic regression model. To simulate a natural heterogeneous setting, the data are distributed among 1,000 devices. For convenience, each device only owns samples of two types of digits, and the number of samples per device follows a power law.
- FEMNIST: We then perform a more complex image classification task on the 62-class Federated Extended MNIST [50] with the same model. Ten lower case characters, including 'a' to 'j', are sampled to generate heterogeneous data partitions, and only five classes are distributed to all 200 devices.

The experiments are implemented in TensorFlow, simulating a federated network with one parameter server and K clients,

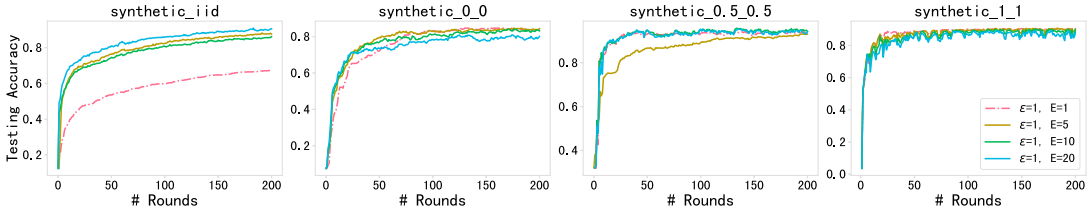


Fig. 5. Testing accuracy with varied E in Synthetic data set.

where K is the total number of clients in the data sets. For each experiment, we select partial clients to participate in the FL process and set $n(t)$ as ten clients. The study is based on an image classification task with MNIST using a multinomial logistic regression model. Firstly, we generate one IID data set and three Non-IID data sets. For all the Synthetic data sets, there are 100 clients in total, and the number of data on each client follows a power law, which constructs the statistical heterogeneity in the experiments. Finally, we study a more real classification task with FEMNIST, a 62-class Federated Extended MNIST data set with the same model.

B. Customized Local Training

To better understand how the customized local training strategy works in vehicular networks, we explore the impact of statistical heterogeneity on convergence speed.

First, we demonstrate how statistical heterogeneity affects convergence with four synthetic data sets, as shown in Fig. 6. The parameters α and β in synthetic_α_β denote the statistical heterogeneity degree of the data sets, which increases gradually from left to right. The higher values of α and β represent the higher statistical heterogeneity. Besides, the total communication rounds are fixed as 200 rounds. The critical parameters of FedCPF that affect performance are the amount of local computing on the clients and the parameter ε in the customized local training strategy. From the perspective of the FL process, choosing a larger amount of local computing reduces the overall communication rounds T, thereby reducing communication costs. However, increasing the amount of local computing will introduce noises of the system. In general, large E may cause local models to drift too far away from the initial beginning point, leading to potential updating divergence. Therefore, a constraint item to reduce the distance between local models and the global model is proposed in Sec. IV-C. A small amount of local computing will cause local models to converge to the local optimal solutions instead of the global optimal solution. Theoretically, we are not inclined to choose a small value of E. Based on the experiment results in Fig. 5, the accuracy of FedCPF with varied E settings is similar, especially when the E set as 10 and 20. However, when each device runs at most one epoch at each iteration ($E = 1$), the accuracy of experiments is a bit different. Hence, to make up for the difference caused by the distribution of the Non-IID data set, it is necessary to set E reasonably. We eliminate the effects of system heterogeneity by forcing each client to run the same amount of local computing, considering that the primary purpose of our work is to improve communication efficiency and save communication overheads. Therefore, a trade-off is made between local computing and global communication overheads. Then the local training epoch is fixed as $E = 20$ in this section. For four synthetic data sets, ε is set as 0, 0.3, 0.8, and 1 to denote the varied degree of penalization, respectively. When $\varepsilon = 0$, it represents the FL setting in FedAvg with various statistical heterogeneity degrees.

We compare the accuracy and loss function values of our proposed approach FedCPF (with varied values of ε) to the baseline approach FedCPF (with $\varepsilon = 0$), respectively. In Fig. 5(a), the convergence speed becomes slow for FedCPF with $\varepsilon = 0$, as the pink curve depicted. When $\varepsilon = 0.3$ and $\varepsilon = 0.8$, the green and orange curves both outperform the baseline, respectively. However, with the statistical heterogeneity raises, the oscillation amplitude of the two curves is more severe. Although both of them eventually show a trend of convergence, the accuracy of both is lower than the blue curve. Under various parameter settings, the accuracy decreases with the increase of statistical heterogeneity. Considering the overall performance in all results, the blue curve shows the most stable and convergent trend in all settings of ε. For IID data, it may slow convergence, but setting with $\varepsilon = 1$ and fixed E have a pronounced effect in heterogeneous networks. The same trend is reflected in the curve of loss function value in Fig. 6(b).

As equation (7) depicts, the constraint item measures the distances between local models and the global model after each communication round. The constraint item penalizes local models that deviate too much from the global model. The coefficient ε is used to measure the weight of penalization in the entire loss function $g_k\left(\boldsymbol{\omega}; \boldsymbol{\omega}^t\right)$. Increasing statistical heterogeneity leads to worse convergence, but the constraint item alleviates this issue. The experimental results show that when the coefficient of penalization has a greater weight, a larger ε value has a better and stable performance in varied situations. Consequently, ε is set as 1 in subsequent experiments based on the above results. Based on the experimental results, the customized local training strategy introduced in FedCPF is beneficial for the real federated settings with varied statistical heterogeneity. The changing trends of loss function values are consistent with the accuracy, as shown in Fig. 6(b). Consequently, the total communication rounds T are decreased by FedCPF, and the total communication costs is reduced.

From experimental results in Fig. 6 with the customized local training strategy, we easily access the newly updated global model and then distribute it to the newly joined clients.

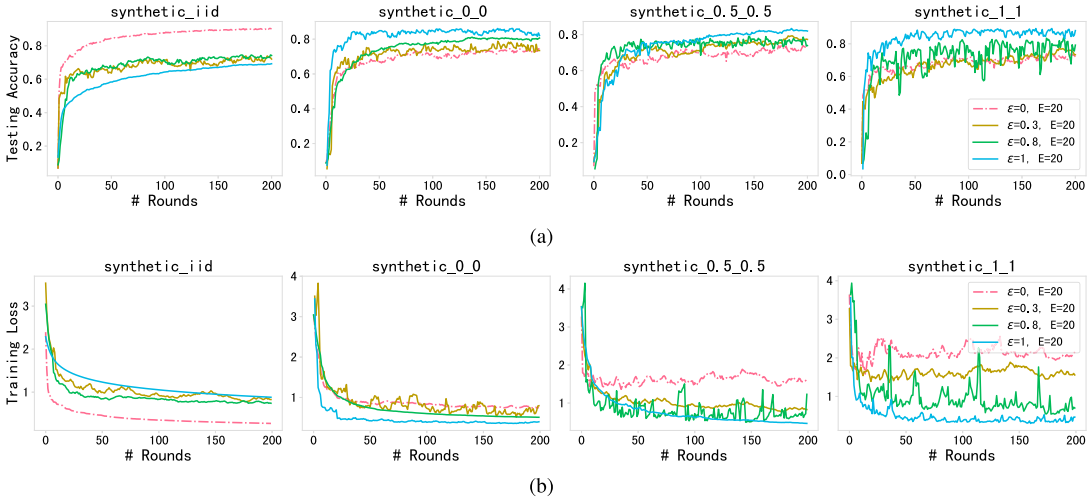


Fig. 6. Full results with varied statistical heterogeneity in Synthetic data set.

To better save training time and communication costs, FedCPF improves the convergence speed on current vehicular networks. Besides, the communication rounds T are significantly decreased by FedCPF compared with FedAvg. Meanwhile, the convergence speed in FedCPF can be guaranteed. For synthetic_0_0, synthetic_0.5_0.5, and synthetic_1_1 data sets, the accuracy in FedCPF increases by 4.72%, 3.69%, and 10.51%, compared with the FedAvg, respectively.

C. Flexible Aggregation

To better measure the effect of performing the flexible aggregation policy to tackle the expensive communication overheads with FedCPF, we simulate the federated settings with fluctuating system heterogeneity.

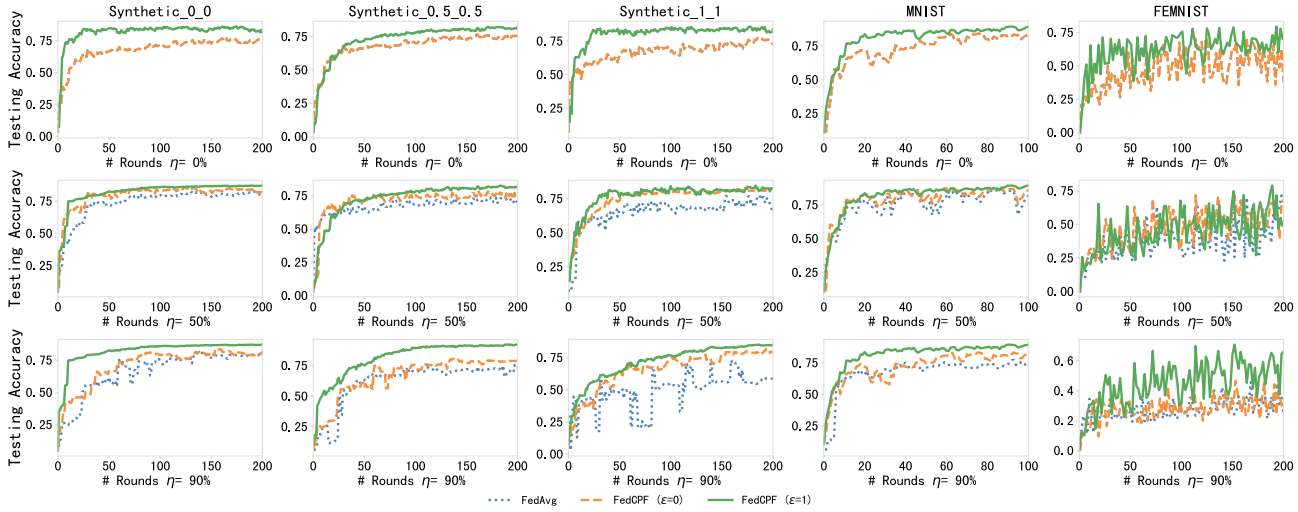
We assume that there exists a total running time during the federated training phase. Each participating client determines the amount of local training as a function of the time and its systems constraints. This specified amount of local computation corresponds to some implicit value θ_k^t for client k at the t-th federated training phase. In our experiments, we fix a global number of epochs E and allow some clients to perform fewer updates than E epochs given the constraints of their current system. For different η settings, at each communication round, we denote η as the number of epochs to 0%, 50%, and 90% of the selected clients. When η is 0%, no client performs fewer epochs than E epochs of work correspond to the environments without system heterogeneity. In contrast, 90% of the clients send their local training outputs to the parameter server. FedCPF drops the 0%, 50%, and 90% inactive clients and then flexible incorporates the customized local training strategy into these clients in the constraint time.

As shown in Fig. 7, the experimental results in the charts show the FL accuracy and training loss on different data sets. In this setting, we find that FedCPF with different values of η guarantee the convergence while FedAvg cannot guarantee this issue. η represents the degree of the system heterogeneity. With the small value of η, the degree of system heterogeneity is lower. From the testing accuracy shown in Fig. 7(a), we find that the increased η intensifies the fluctuation in all the data sets. When $\eta = 0\%$, the experimental results is more stable than $\eta = 90\%$ in all the data sets. The oscillation amplitude of the curve using the FedAvg algorithm becomes more evident as the η increases. Besides, the convergence speed of FedAvg is slower than the FedCPF, which means it needs lots of communication rounds to acquire a good performance. In FedCPF, when $\varepsilon = 0$ and $\eta = 0\%$, the FedCPF degenerates to FedAvg. The curve of FedCPF with $\varepsilon = 0$ is more stable than the FedAvg in all the data sets. Finally, the experimental results show that the convergence speed of FedCPF is faster than FedAvg, even if η is high. The FedCPF has the most stable performance among these three varied settings ($\eta = 0\%, 50\%, 90\%$). Meanwhile, the accuracy of the FedCPF is higher than the FedAvg on all settings. Besides, due to the high complexity of the FEMNIST, the amplitude of the curve oscillation is relatively large. For these five different data sets, the testing accuracy in FedCPF increases by 4.36%, 10.75%, 22.01%, 12.61%, and 33.61%, respectively, compared with the FedAvg. The trend of the training loss value curve in the experiment is opposite to the trend of the testing accuracy curve. When using FedAvg, the loss reduction in Fig. 7(b) is slight compared with the loss of the FedCPF. The curve oscillation amplitude is relatively large, which makes it challenging to show the convergence trend.

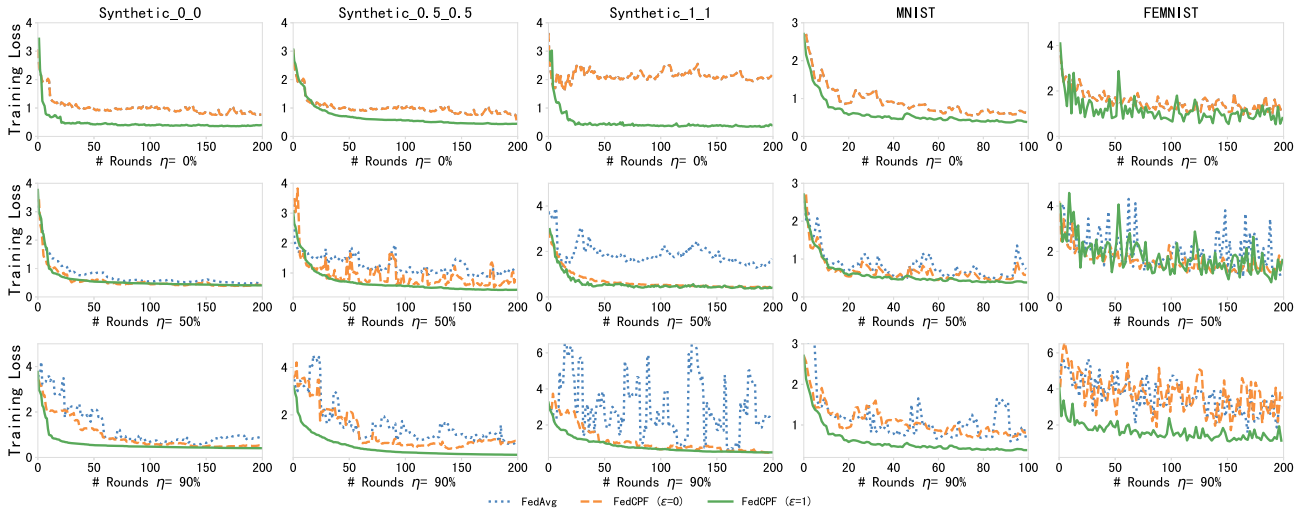
D. Communication Costs

To statistic the communication costs in the whole FL process, we provide the communication results of each client on average on the above data sets in Fig. 8. For the different data sets, the degree of communication optimization will be varied dramatically because of the various neural network structures. The experimental results are shown as follows.

The bar graphs in Fig. 8 show the communication costs of each client on average in one FL communication round with Synthetic, MNIST, FEMNIST, respectively. The parameter of FedAvg, f, represents the fraction of clients who participated



(a)



(b)

Fig. 7. Full results with system heterogeneity on different data sets.

in each FL round, which is set as 0.1. FedCPF($\eta = 90\%$), FedCPF($\eta = 50\%$), and FedCPF($\eta = 0\%$) all benefit from the FedCPF algorithm. Compared with FedAvg, the algorithm achieves communication optimization in these data sets to some extent. For the Synthetic data set, the communication overheads for one communication round of each client is 1.86MB when FedAvg is used. The optimization degree of communication costs in FedCPF($\eta = 90\%$) is the best due to discarding most clients. For the different settings of FedCPF, the communication optimization rates are 0.43%, 12.90%, and 22.85% compared with FedAvg, respectively. For MNIST, the communication overheads of each client are 299.8MB when FedAvg is used. The communication optimization rates are 0.13%, 0.62%, and 1.02% compared with FedAvg, respectively. For FEMNIST, the communication overheads of each client are 327MB when FedAvg is used. The communication optimization rates are 0.50%, 2.87%, and 4.76% compared with FedAvg, respectively. Above all, the communication optimization of FedCPF is effective.

In Table II and Table III, each table entry shows the necessary communication rounds to reach a target testing accuracy for multi-class logistic regression (mclr) models on different data sets. Based on the experimental results, we observe the communication optimization effects of the FedCPF algorithm compared with the baseline of FedAvg.

Table II shows the impact of varying ε, η on three synthetic data sets with varied statistical heterogeneity. When the parameter η exists, it means that the FedCPF algorithm contains a constraint item. The overall communication optimization effect of the FedCPF algorithm is significantly higher than that of the FedAvg algorithm. Compared with FedCPF ($\varepsilon = 0$, $\eta = 50\%$) and FedAvg ($f = 0.1$, $\eta = 0\%$) algorithms, the average communication optimization rate is 1.74 times. When the skewness of data distribution in the Synthetic data set gradually increases, the number of communication rounds required to achieve the same target accuracy gradually rises. When the skewness of data distribution in the Synthetic data set gradually increases, the number of communication rounds

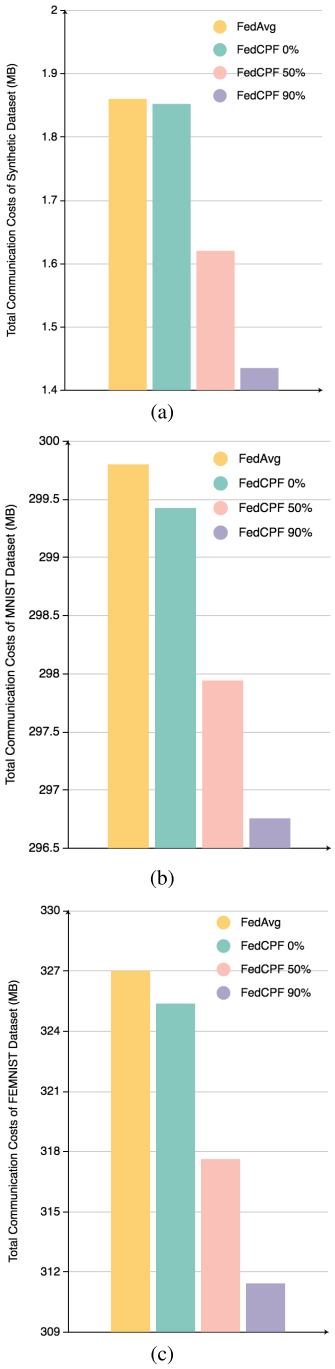


Fig. 8. The communication costs of one client in different data sets with FedAvg and FedCPF.

required to achieve the same target accuracy gradually rises. Considering the skewness of the data distribution determines the statistical heterogeneity degree of the data set. Statistical heterogeneity has always been a challenge for FL algorithms, dramatically increasing the difficulty of convergence in FL algorithms. Consequently, when the statistical heterogeneity degree is low, the communication optimization effects are more significant than the data set with a high statistical heterogeneity degree in all the algorithms. However, in the FedCPF algorithm, the communication optimization effect is still significant, and the optimization effect is improved

TABLE II
EFFECT OF THE COMMUNICATION OPTIMIZATION ON SYNTHETIC DATA SET

Algorithm	Synthetic data set acc=75%		
	_0_0	_0.5_0.5	_1_1
FedAvg ($f = 0.1$, $\eta = 0\%$)	65	114	151
FedCPF ($\varepsilon = 0$, $\eta = 0\%$)	59 (1.0×)	110 (1.0×)	146 (1.0×)
FedCPF ($\varepsilon = 1$, $\eta = 0\%$)	15 (4.3×)	59 (1.9×)	85 (1.8×)
FedAvg ($f = 0.1$, $\eta = 50\%$)	69	143	170
FedCPF ($\varepsilon = 0$, $\eta = 50\%$)	62 (1.2×)	126 (1.1×)	162 (1.0×)
FedCPF ($\varepsilon = 1$, $\eta = 50\%$)	22 (3.1×)	62 (2.3×)	138 (1.2×)
FedAvg ($f = 0.1$, $\eta = 90\%$)	96	184	189
FedCPF ($\varepsilon = 0$, $\eta = 90\%$)	68 (1.4×)	166 (1.1×)	176 (1.0×)
FedCPF ($\varepsilon = 1$, $\eta = 90\%$)	26 (3.7×)	65 (2.4×)	163 (1.2×)

TABLE III
EFFECT OF THE COMMUNICATION OPTIMIZATION ON MNIST AND FEMNIST DATA SETS

Algorithm	MNIST acc=75%	FEMNIST acc=60%
FedAvg ($f = 0.1$, $\eta = 0\%$)	46	103
FedCPF ($\varepsilon = 0$, $\eta = 0\%$)	42 (1.1×)	86 (1.2×)
FedCPF ($\varepsilon = 1$, $\eta = 0\%$)	11 (4.2×)	69 (1.5×)
FedAvg ($f = 0.1$, $\eta = 50\%$)	55	164
FedCPF ($\varepsilon = 0$, $\eta = 50\%$)	45 (1.2×)	159 (1.0×)
FedCPF ($\varepsilon = 1$, $\eta = 50\%$)	11 (5.0×)	135 (1.2×)
FedAvg ($f = 0.1$, $\eta = 90\%$)	59	–
FedCPF ($\varepsilon = 0$, $\eta = 90\%$)	45 (1.3×)	– (–)
FedCPF ($\varepsilon = 1$, $\eta = 90\%$)	13 (4.5×)	124 (–)

by 2.35 times. As the proportion of straggler clients rises, so does the system heterogeneity. The convergence speed of the FedAvg algorithm decreases significantly, and the number of communication rounds required gradually increases. Compared with FedAvg (the parameter f is fixed), FedCPF still has a communication optimization effect under the condition of increased system heterogeneity.

Table III shows the impact of varying ε, η on MNIST and FEMNIST data sets with accuracy 75% and 60%, respectively. For MNIST, the average communication optimization rate is improved by 2.9 times. For FEMNIST, the average communication optimization rate is improved by 1.2 times. Since FEMNIST has many labels, the difficulty of processing the classification task will also increase. There are '-' represents the unreachable target accuracy within the allowed communication rounds. Above all, the communication optimization of FedCPF is obvious.

VI. CONCLUSION

In this work, we have proposed FedCPF, an efficient-communication approach that tackles the expensive communication costs in vehicular networks. FedCPF allows for customized training across varied clients and increases the convergence speed. The customized local training strategy is proposed for fewer communication rounds to improve communication efficiency. The partial client participation rule allows a few clients to upload simultaneously to decrease communication costs in every round. The flexible aggregation policy allows the straggler clients during the aggregation phase to dynamically adjust the number of clients to reduce each communication overheads further. Our experimental results show that across different heterogeneity data sets, FedCPF has

a convergence guarantee. The FedCPF approach significantly improves the convergence speed and decrease the communication costs in vehicular networks.

APPENDIX A
PROOF OF THE CONVERGENCE ANALYSIS

Through the bounded diversity in Assumption 1, we analyze the expected reduction in the objective functions when performing one FedCPF epoch. In our work, the convergence speed can be directly derived from the results of the expected reduction per training round. For convenience, suppose that the same θ_k^t for any k, t in the following analyses.

A. Non-Convex FedCPF Convergence: V-Local Diversity

We assume the F_k is a *non-convex*, *L-Lipschitz* and *smooth* function. There exists $L_- > 0$, such that $\nabla^2 F_k \succeq -L_-\mathrm{I}$, with $\bar{\varepsilon} := \varepsilon - L_- > 0$. Suppose that ω^t is not a stationary solution and the local functions F_k are V-diversity, i.e. $V(\omega^t) \leq V$. If ε, K and θ in Algorithm 1 are chosen such as equation (14):

$$\varphi = \left(\frac{1}{\varepsilon} - \frac{\theta V}{\varepsilon} - \frac{V(1+\theta)\sqrt{2}}{\bar{\varepsilon}\sqrt{K}} - \frac{LV(1+\theta)}{\bar{\varepsilon}\varepsilon} - \frac{L(1+\theta)^2V^2}{2\bar{\varepsilon}^2} - \frac{LV^2(1+\theta)^2}{\bar{\theta}^2 K}(2\sqrt{2K}+2)\right) > 0. \quad (14)$$

then at the t-th iteration of algorithm 1, we have the following expected reduction in the global objective function:

$$\mathbb{E}_{S_t}[f(\omega^{t+1})] \leq f(\omega^t) - \varphi \left\|\nabla f(\omega^t)\right\|^2. \quad (15)$$

where S_t is the set of K devices chosen at iteration t. The critical steps include applying the notion of θ-inexactness solution for each subproblem and using the bounded diversity assumption. This process allows for only K devices to be active at the same round. In particular, an expectation $\mathbb{E}_{S_t}$ with respect is introduced to the selection of devices S_t in round t. We note that in our work, we require $\bar{\theta} > 0$, which is a sufficient but not necessary condition for FedCPF to converge. Hence, it is possible that some other θ, maybe not necessarily satisfy $\bar{\theta} > 0$), can also enable convergence, as we explore empirically.

B. Convex FedCPF Convergence

Let the above situation is hold. Based on the above discussion, let $F_k(\cdot)$'s be convex and $\theta_k^t = 0$ for any k, t, i.e., all the local functions are solved exactly, if $1 \ll V \leq \frac{1}{2}\sqrt{K}$, then we choose $\varepsilon \approx 6LV^2$ from which it follows that $\varphi \approx \frac{1}{24LV^2}$.

Above all, the analysis is helpful to characterize the performance of FedCPF and similar methods when local functions are dissimilar.

ACKNOWLEDGMENT

The authors are grateful to the anonymous referees for their insightful suggestions and comments.

REFERENCES

[1] Z. Lu, G. Qu, and Z. Liu, "A survey on recent advances in vehicular network security, trust, and privacy," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 2, pp. 760–776, Feb. 2019.
[2] W. Qi, Q. Li, Q. Song, L. Guo, and A. Jamalipour, "Extensive edge intelligence for future vehicular networks in 6G," *IEEE Wireless Commun.*, early access, Mar. 19, 2021, doi: 10.1109/MWC.001.2000393.
[3] K. M. S. Huq, S. Mumtaz, J. Rodriguez, P. Marques, B. Okyere, and V. Frascolla, "Enhanced C-RAN using D2D network," *IEEE Commun. Mag.*, vol. 55, no. 3, pp. 100–107, Mar. 2017.
[4] S. Mumtaz, H. Lundqvist, K. M. S. Huq, J. Rodriguez, and A. Radwan, "Smart direct-LTE communication: An energy saving perspective," *Ad Hoc Netw.*, vol. 13, pp. 296–311, Feb. 2014.
[5] W. Chen, X. Qiu, T. Cai, H.-N. Dai, Z. Zheng, and Y. Zhang, "Deep reinforcement learning for Internet of Things: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, early access, Apr. 13, 2021, doi: 10.1109/COMST.2021.3073036.
[6] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 869–904, 2nd Quart., 2020.
[7] X. Deng, F. Long, B. Li, D. Cao, and Y. Pan, "An influence model based on heterogeneous online social network for influence maximization," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 2, pp. 737–749, Apr. 2020.
[8] B. Li, X. Deng, and Y. Deng, "Mobile-edge computing-based delay minimization controller placement in SDN-IoV," *Comput. Netw.*, vol. 193, Jul. 2021, Art. no. 108049.
[9] X. Deng, Y. Liu, C. Zhu, and H. Zhang, "Air–ground surveillance sensor network based on edge computing for target tracking," *Comput. Commun.*, vol. 166, pp. 254–261, Jan. 2021.
[10] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, 2019.
[11] T. Wang *et al.*, "Privacy-enhanced data collection based on deep learning for internet of vehicles," *IEEE Trans. Ind. Informat.*, vol. 16, no. 10, pp. 6663–6672, Oct. 2020.
[12] Y. Liu, J. J. Q. Yu, J. Kang, D. Niyato, and S. Zhang, "Privacy-preserving traffic flow prediction: A federated learning approach," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7751–7763, Aug. 2020.
[13] C. Chen, B. Liu, S. Wan, P. Qiao, and Q. Pei, "An edge traffic flow detection scheme based on deep learning in an intelligent transportation system," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1840–1852, Mar. 2021.
[14] J. Jin, H. Guo, J. Xu, X. Wang, and F.-Y. Wang, "An end-to-end recommendation system for urban traffic controls and management under a parallel learning framework," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1616–1626, Mar. 2021.
[15] G. P. Corser, H. Fu, and A. Banihani, "Evaluating location privacy in vehicular communications and applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 9, pp. 2658–2667, Sep. 2016.
[16] J. Posner, L. Tseng, M. Aloqaily, and Y. Jararweh, "Federated learning in vehicular networks: Opportunities and solutions," *IEEE Netw.*, vol. 35, no. 2, pp. 152–159, Mar. 2021.
[17] M. Z. Khan, S. Harous, S. U. Hassan, M. U. G. Khan, R. Iqbal, and S. Mumtaz, "Deep unified model for face recognition based on convolution neural network and edge computing," *IEEE Access*, vol. 7, pp. 72622–72633, 2019.
[18] L.-L. Wang, J.-S. Gui, X.-H. Deng, F. Zeng, and Z.-F. Kuang, "Routing algorithm based on vehicle position analysis for internet of vehicles," *IEEE Internet Things J.*, vol. 7, no. 12, pp. 11701–11712, Dec. 2020.
[19] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big data analytics in intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 1, pp. 383–398, Jan. 2019.
[20] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artif. Intell. Statist.*, 2017, pp. 1273–1282.
[21] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst.*, vol. 2, 2020, pp. 429–450.
[22] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Convergence time optimization for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 4, pp. 2457–2471, Apr. 2021.
[23] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, "FedPAQ: A communication-efficient federated learning method with periodic averaging and quantization," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2019, pp. 2021–2031.

[24] L. Li, H. Xiong, Z. Guo, J. Wang, and C.-Z. Xu, "SmartPC: Hierarchical pace control in real-time federated learning system," in *Proc. IEEE Real-Time Syst. Symp. (RTSS)*, Dec. 2019, pp. 406–418.

[25] D. Ye, R. Yu, M. Pan, and Z. Han, "Federated learning in vehicular edge computing: A selective model aggregation approach," *IEEE Access*, vol. 8, pp. 23920–23935, 2020.

[26] H. Yu *et al.*, "A fairness-aware incentive scheme for federated learning," in *Proc. AAAI/ACM Conf. AI, Ethics, Soc.*, Feb. 2020, pp. 393–399.

[27] S. Dhakal, S. Prakash, Y. Yona, S. Talwar, and N. Himayat, "Coded federated learning," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2019, pp. 1–6.

[28] S. Prakash *et al.*, "Coded computing for low-latency federated learning over wireless edge networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 233–250, Jan. 2021.

[29] S. Wang *et al.*, "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 3, pp. 1205–1221, Jun. 2019.

[30] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[31] X. Yao, T. Huang, C. Wu, R. Zhang, and L. Sun, "Federated learning with additional mechanisms on clients to reduce communication costs," *CoRR*, vol. abs/1908.05891, Sep. 2019. [Online]. Available: http://arxiv.org/abs/1908.05891

[32] H. T. Nguyen, V. Sehwag, S. Hosseinalipour, C. G. Brinton, M. Chiang, and H. V. Poor, "Fast-convergent federated learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 201–218, Jan. 2021.

[33] S. Li, Q. Qi, J. Wang, H. Sun, Y. Li, and F. R. Yu, "GGS: General gradient sparsification for federated learning in edge computing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–7.

[34] N. Shlezinger, M. Chen, Y. C. Eldar, H. V. Poor, and S. Cui, "UVeQFed: Universal vector quantization for federated learning," *IEEE Trans. Signal Process.*, vol. 69, pp. 500–514, Jan. 2021.

[35] M. Li, Y. Chen, Y. Wang, and Y. Pan, "Efficient asynchronous vertical federated learning via gradient prediction and double-end sparse compression," in *Proc. 16th Int. Conf. Control, Automat., Robot. Vis. (ICARCV)*, Dec. 2020, pp. 291–296.

[36] H. Zhou, J. Cheng, X. Wang, and B. Jin, "Low rank communication for federated learning," in *Proc. Int. Conf. Database Syst. Adv. Appl.*, 2020, pp. 1–16.

[37] L. Wang, W. Wang, and B. Li, "CMFL: Mitigating communication overhead for federated learning," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2019, pp. 954–964.

[38] H. Yang, M. Fang, and J. Liu, "Achieving linear speedup with partial worker participation in non-IID federated learning," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–23.

[39] Y. Ruan, X. Zhang, S.-C. Liang, and C. Joe-Wong, "Towards flexible device participation in federated learning for non-IID data," 2020, *arXiv:2006.06954*. [Online]. Available: https://arxiv.org/abs/2006.06954

[40] W. Zhuang *et al.*, "Performance optimization of federated person re-identification via benchmark analysis," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 955–963.

[41] K. Bonawitz *et al.*, "Towards federated learning at scale: System design," in *Proc. Mach. Learn. Syst.*, vol. 1, 2019, pp. 374–388.

[42] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Distributed federated learning for ultra-reliable low-latency vehicular communications," *IEEE Trans. Commun.*, vol. 68, no. 2, pp. 1146–1159, Nov. 2019.

[43] J. Ren, H. Wang, T. Hou, S. Zheng, and C. Tang, "Federated learning-based computation offloading optimization in edge computing-supported Internet of Things," *IEEE Access*, vol. 7, pp. 69194–69201, 2019.

[44] C. Chen, Y. Zhang, Z. Wang, S. Wan, and Q. Pei, "Distributed computation offloading method based on deep reinforcement learning in ICV," *Appl. Soft Comput.*, vol. 103, May 2021, Art. no. 107108.

[45] X. Deng, Z. Sun, D. Li, J. Luo, and S. Wan, "User-centric computation offloading for edge computing," *IEEE Internet Things J.*, early access, Feb. 8, 2021, doi: 10.1109/JIOT.2021.3057694.

[46] L. Liu, J. Zhang, S. H. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6.

[47] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-IID data," 2018, *arXiv:1806.00582*. [Online]. Available: http://arxiv.org/abs/1806.00582

[48] S. Caldas *et al.*, "LEAF: A benchmark for federated settings," 2018, *arXiv:1812.01097*. [Online]. Available: http://arxiv.org/abs/1812.01097

[49] O. Shamir, N. Srebro, and T. Zhang, "Communication-efficient distributed optimization using an approximate Newton-type method," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, pp. 1000–1008.

[50] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, "EMNIST: An extension of MNIST to handwritten letters," in *Proc. Comput. Vis. Pattern Recognit.*, 2017.

Su Liu received the B.E. degree from the Chengdu University of Technology in 2016 and the M.E. degree from Xinjiang University, China, in 2019, where she is currently pursuing the Ph.D. degree with the School of Information Science and Engineering. Her current research interests include federated learning, edge computing, and distributed computing.

Jiong Yu received the Ph.D. degree from the School of Computer Science and Technology, Beijing University of Technology, China, in 2009. He worked as a Senior Visiting Scholar with the Information Computing Center, National Research Institute, Canada. He is currently a Professor and the Ph.D. Supervisor in computer science with the School of Information Science and Engineering, Xinjiang University. His main researches focus on grid computing, parallel computing, and deep learning.

Xiaoheng Deng (Member, IEEE) received the Ph.D. degree in computer science from Central South University, Changsha, Hunan, China, in 2005. Since 2006, he has been an Associate Professor and then a Full Professor with School of Computer Science and Engineering, Central South University. He is currently the Chair of RS Changsha Chapter, a Senior Member of CCF, and a member of CCF Pervasive Computing Council and ACM. He was the Chair of CCF YOCSEF CHANG SHA from 2009 to 2010. His research interests include wireless communications and networking, edge computing, congestion control for wired/wireless networks, cross layer route design for wireless mesh networks and *ad hoc* networks, online social network analysis, and distributed computing systems.

Shaohua Wan (Senior Member, IEEE) received the Ph.D. degree from the School of Computer, Wuhan University in 2010. Since 2015, he has been holding a post-doctoral position with the State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology. From 2016 to 2017, he was a Visiting Professor with the Department of Electrical and Computer Engineering, Technical University of Munich, Germany. He is currently an Associate Professor with the School of Information and Safety Engineering, Zhongnan University of Economics and Law. His main research interests include deep learning for the Internet of Things and edge computing. He is an author of over 110 peer-reviewed SCI indexed papers. He had served as the lead Guest Editor of IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, *ACM Transactions on Multimedia Computing Communication*, *Journal of Systems Architecture*, *Computer Communications*, *Pattern Recognition Letter*, *Multimedia Tools and Applications*, *Image and Vision Computing, and Computers and Electrical Engineering*.