Semi-Synchronous Federated Learning Protocol With Dynamic Aggregation in Internet of Vehicles

Feiyuan Liang, Qinglin Yang, *Member, IEEE*, Ruiqi Liu, Junbo Wang, *Member, IEEE*, Kento Sato, *Member, IEEE*, and Jian Guo, *Member, IEEE*

***Abstract*—In an Internet of Vehicle (IoV) system, federated learning (FL) is a new approach to process real-time vehicle data in a distributed way, which can improve the driving experience and service quality. However, due to the high mobility and uncertainty of vehicles, the existing federated learning protocols are difficult to meet the full requirements from an IoV system, such as efficient resource allocation, high precision learning, and fast convergence of learning algorithm. To solve the above problems, in this paper, we propose a semi-synchronous federated learning (Semi-SynFed) protocol, to improve the performance of machine learning at Internet of Vehicles. In Semi-SynFed, we first select appropriate nodes to participate the aggregation by its computing capacity, network capacity and learning value of training samples. Meanwhile, the dynamic waiting time technique is designed to adjust the server waiting time at each round dynamically, which makes federated learning process work much more efficient. Finally, a dynamic aggregation scheme is designed to aggregate model parameters in an asynchronous way. To evaluate the reliability of Semi-SynFed, we establish a simulation of IoVs based on the real world data. In the comparative experiments, the results show that the Semi-SynFed outperforms the existing federated learning protocols in terms of convergence speed and resource consumption.**

***Index Terms*—Dynamic aggregation, federated learning, intelligent connected vehicles, Internet of Vehicles, semi-synchronization.**

I. Introduction

INTELLIGENT transportation system (ITS) has been developed since the 1970s [1], trying to solve traffic problems (e.g., traffic congestion) by sensing, data collecting, and data processing. The data can be collected by a large number of roadside sensors, such as RF readers, cameras, even from social media that can be used to provide a more convenient service to users by detecting traffic accidents [2], [3], predicting traffic flows [4] and performing schedule planning for public transportation services [5].

In most recent years, with the development of sensing and data processing technologies at edge devices, the Internet of Vehicles (IoV) and Intelligent Connected Vehicle (ICV) play a more and more important effort in ITS. Many researchers are focusing on how to process the data collected by vehicles in a real-time manner. Machine learning models are often built on the data collected by on-board sensors at ICV. The data contains sufficient information (e.g., vehicle status, driving behavior, traffic environment) that can support situation understanding and the prediction of the future events under complex traffic scenarios [6]–[8].

The conventional methods for machine learning often require vehicle to send data to the data center for centralized model training through IoV. Although there are several advantages and many applications of the centralized model training, it still meets some obstacles to in-depth utilize the real-time vehicle data, due to the following reasons: First, ICV can generate the data by radar and camera at the rate of 30.23 $GB \cdot h^{-1}$ [9], which brings huge burden to the limit communication network especially when it comes to the communication condition of IoV. In addition, the data generated by distributed devices owned by individuals or organization, they may be hesitate to transfer their sensitive data that expose privacy to the cloud.

Federated Learning (FL) was proposed to solve the aforementioned problems, which is a technological framework that allows the machine learning model to learn knowledge from the distributed repositories owned by different organizations or devices without raw data transferring. In FL, the computing nodes train local models by using their own data, respectively, and then upload the local model instead of data, to a logical centralized parameter server that aggregates all local models into a global model [10].

Recently, more and more studies tried to deploy FL framework to IoV applications [11], [12]. The authors in [11] proposed FedParking using LSTM model to realize parking space estimation with privacy protection. In [12], the authors developed a novel dynamic map fusion scheme among intelligent networked vehicles based on federated learning.

The existing works on FL protocols are divided into two aspects: synchronous [13], and asynchronous methods [14], according to different types of aggregation. For the synchronous FL protocol, the parameter server needs to collect all the

Manuscript received June 7, 2021; revised October 20, 2021 and December 24, 2021; accepted January 17, 2022. Date of publication February 7, 2022; date of current version May 20, 2022. This work was supported by the National Nature Science Foundation of China under Grant 62072485. The review of this article was coordinated by Dr. Xuanyu Cao. *(Corresponding author: Junbo Wang.)*

Feiyuan Liang, Qinglin Yang, and Ruiqi Liu are with the Sun Yat-Sen University, Guangzhou, Guangdong 510275, China (e-mail: liangfy9@mail2.sysu.edu.cn; yangqlin6@mail.sysu.edu.cn; liurq9@mail2.sysu.edu.cn).

Junbo Wang is with the Sun Yat-Sen University, Guangzhou, Guangdong 510275, China, and also with the School of Intelligent Systems Engineering and Guangdong Provincial Key Laboratory of Intelligent Transportation System, Sun Yat-Sen University, Guangzhou, Guangdong 510275, China (e-mail: wangjb33@mail.sysu.edu.cn).

Kento Sato is with Riken Kobe Branch, Kobe, Hyogo 351-0198, Japan (e-mail: kento.sato@riken.jp).

Jian Guo is with the Xi'an University of Finance and Economics, Xi'an, Shaanxi 710064, China (e-mail: jian_guo@xaufe.edu.cn).

Digital Object Identifier 10.1109/TVT.2022.3148872

parameters from the clients before executing aggregation procedure, which will bring the straggler issue due to the heterogeneity of client's computing power and communication. On the contrary, in an asynchronous FL protocol, the parameter server can aggregate the parameters without waiting all clients in one round, which means the number of participating clients at each round is uncontrollable.

Even the existing synchronous and asynchronous methods work well in some FL scenarios, but it still remains several challenges in IoV as listed following:

- The data is distributed heterogeneously among many vehicular nodes, and thus the bias of local data can greatly affect the accuracy of the global model;
- Vehicles usually move very fast, and the computation and communication resources of vehicular nodes are diverse and vary over time based on the distance with a base station. Besides, some vehicles may be disconnected within an uncertain time, which will greatly affect the global model's convergence.

To solve the aforementioned challenges, we propose a novel semi-synchronous FL (Semi-SynFed) protocol for IoV systems. We firstly design a client selection scheme to evaluate the status of vehicular nodes. The vehicular nodes satisfying upper limit of communication rounds will participate in the aggregation synchronously. Meanwhile, a server waiting time scheme is designed to adjust server aggregation frequency dynamically. To evaluate the proposed method, a simulation environment of IoV based on the real road network and the base station (BS) location data. Compared with the other three FL protocols at four machine learning tasks, the proposed protocol performs better in convergence speed, accuracy, and resource cost. The main contributions are summarized as following:

- We design a client selection scheme by considering computing capacity, network capacity, and gradient norm of vehicular nodes.
- We propose Semi-SynFed protocol, which aggregates local models of different generations through synchronous methods. In addition, Semi-SynFed can reduce resource and communication costs by adjusting server waiting time.
- We establish a simulation of IoV based on the road network and BS location data of Guangzhou Higher Education Mega Center. Machine learning task experiments show that our algorithm performs well in accuracy, convergence speed, and resource cost under fixed system runtime.

The rest of this paper is organized as following. The related works of federated learning are summarized in Section II. Section III describes the scenario and optimization problem of FL under IoV. Section IV provides the details of the node selection scheme and Semi-SynFed protocol. Section V presents the simulation of IoV and experimental evaluation of the proposed protocol at four machine learning tasks. Finally, Section VI concludes this work.

II. Related Work

The current FL protocols can be mainly divided into synchronous methods and asynchronous methods by aggregation types.

A. Synchronization

The idea of synchronous methods is that parameter server waits for all or most of nodes to complete local training and uploading in each global round, and then performs aggregation and broadcasts. FedAvg is a typical synchronous FL protocol [13], by which parameter server randomly selects model parameters at a fixed ratio to compute the weighted average of parameters once all nodes upload their local parameters at each round. Global models only need several global rounds to achieve high accuracy under FedAvg protocol.

In terms of improving convergence speed, the authors [15] propose FSVRG, in which the parameter server collects the local gradient from every client and computes the full gradient that can accelerate the convergence of local models in ends at local update. In [16], the authors propose a Semi-Asynchronous FL Protocol (SAFA). In SAFA, a caching scheme is designed to cache model parameters of the nodes that do not be involved in aggregation at current round. Cached parameters can wait for the next aggregation by setting up a fixed server waiting time. For classification problems, the authors [17] propose FL protocol based class-weighted aggregation (FedCA), in which the parameters of classification layer are used as weights for parameter aggregation, instead of using the number of data as weights like FedAvg.

In terms of reducing impact of bad nodes, client selection scheme is usually implemented before server broadcast global model [18]. In client selection, training and update time rate are usually considered as basis [18]. In reputation scheme [20]–[22], parameter server gives node rewards based on the performance in current training task. Server considers reputation value for client selection, so as consider to historical nodes' performance.

However, most of the existing synchronous methods only consider the communication efficiency without running time. This will cause the model converge slowly when meeting the mobile networks with drastic changes in transmission rate.

B. Asynchronization

The idea of asynchronous methods is that parameter server immediately aggregates local model parameters when a node finishes uploading. The difference with synchronous method is that asynchronous method updates local model parameters to global model individually, while synchronous method updates global model uniformly after all local model parameters are merged. FedAsync is a typical asynchronous FL protocol [23], in which server updates global model according to the uploaded model parameters and staleness of nodes.

To improve communication efficiency, the authors [25] propose activity and resource-aware federated learning protocol (FedAR) that reduces the straggler effect on asynchronous FL process by assigning a trust score to each FL client. In [26], the authors propose temporally weighted aggregation and layerwise asynchronous processing strategy is applied in FL process, thereby communication cost can be reduced.

To improve the convergence accuracy, keeping the balance of each nodes' contribution to global model is a common method [27], [28]. In [27], the number of local epochs is

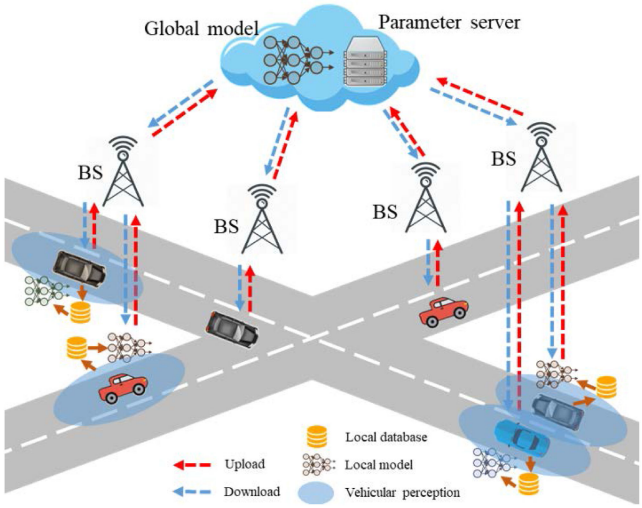


Fig. 1. Illustration of FL process in IoV system.

dynamically adjusted according to the estimation of staleness. Experiment shows that the proposed asynchronous method can converge faster than FedAsync on the existence of massive stragglers. To reduce the impact of overactive nodes, the authors [28] set an age filter to control the discrepancies between global model and delayed model, and demonstrate that CO-OP can make model converge to the expected accuracy quickly.

Although the aforementioned methods have a good performance in the trade-off of convergence speed and accuracy, it is a common issue for rarely considering the limited resource of computation and short and drastic changes in communication rate.

Different from the existing works, we integrate the advantages of synchronous and asynchronous methods to perform semi-synchronous aggregations. Meanwhile, we dynamically adjust maximum server waiting time according to the proportion of participating nodes in each round, allowing as many nodes as possible to participate in the aggregation, thereby reducing communication cost.

III. System Model

In this section, we mainly introduce the federated learning process in the Internet of Vehicles environment, as well as the formulations of problems.

A. Federated Learning in IoV

As depicted in Fig. 1, the system model of the FL consists of intelligent vehicles, base stations (BSs), and cloud server in an IoV environment. We assume there are M ICVs in the road network, and each ICV is equipped with onboard sensors to collect the data (e.g., the status of the vehicle, driving behavior, traffic information) from the surrounding environment. Thereafter, these vehicles collaboratively learn a task via local dataset without raw data transfer. In each learning round, the selected vehicles j download the training model parameter from the parameter server and update the model w_j with the local loss $\ell(w_j)$ according to the local dataset D_j. Subsequently, the selected vehicles upload the new model parameter to the parameter server that aggregates these local models to produce a new global model w^{t+1}. The learning process is repeated until the model reaches the expected accuracy. In addition, some related notations are summarized in Table I.

TABLE I

Notation	Description
B	channel bandwidth
r_k	communication rate at time slot k
P_S	transmitting power of devices
d_k	distance between vehicle and BS
N	spatial noise
η_0	propagation factor
$\mathcal{L}(w)$	global loss function
$\ell(w_j)$	local loss function of vehicular node j
M	number of participating vehicles
n	total sample number of all nodes
n^t	total sample number of participating nodes at global round t
n_j	number of samples of vehicular node j
τ_j^t	total cost time of vehicular node j
ρ_j	device performance of vehicular node j
f_j^t	computation resource of vehicular node j
τ_s^t	server waiting time at global round t
R	total resource waste
T	total communication rounds
N_{ab}^t	abandon nodes set at global round t
S^t	actual participating node set at global round t
w_j^t	model parameters of vehicular node j
w^t	global model parameters at global round t
t_g	global round of server
t_j	local round of vehicular node j
ar^t	actual proportion of participating nodes
ar^e	expected proportion of participating nodes

Let B denote the channel bandwidth between the ICV and BSs. According to Shannon's Capacity Theorem [29], the communication rate r_k at time slot k can be calculated as:

$$r_k = B \cdot \log_2\left(1 + \frac{S}{N}\right), \tag{1}$$

where S denotes the average power of transmitted signals in channel and N denotes the power of Gaussian noise in the inner channel.

For the sake of simplified description, we assume that the communication of IoV is carried out in an open space. Hence, the Signal-Noise ratio $\frac{S}{N}$ can be expressed as

$$\frac{S}{N} = \frac{P_0}{N \cdot d_k^2 \cdot \eta_0}, \tag{2}$$

where d_k is the relative distance between the vehicle and the base station at time slot k, and η_0 is the propagation factor.

Therefore, the communication rate can be calculated as:

$$r_k = B \cdot \log_2\left(1 + \frac{\mathrm{P}_0}{N \cdot d_k^2 \cdot \eta_0}\right). \tag{3}$$

B. Problem Formulation

The objective of federated learning is to learn the model parameters w with the collaboration by a large number of vehicles. The FL learning time includes not only the computation time

(which relies on the hardware types and local data size), but also the communication time of all the vehicles (which rely on the dynamic network and the upload data size).

Time cost: For a single vehicular node j, each global round of FL process can be composed of communication (uploading and downloading) and computation. Hence, the total time τ_j^t consumed in each round can be expressed as

$$\tau_j^t = \tau_{j,c}^t + \tau_{j,com}^t, \tag{4}$$

where $\tau_{j,c}^t$ and $\tau_{j,com}^t$ denote computation time and communication time of node j at global round t, respectively.

To reduce the computational complexity, it can be considered that communication time $\tau_{j,com}^t$ are only related to the model size ϕ and communication rate, while computation time $\tau_{j,c}^t$ is related to the current computing resource f_j^t, the device performance ρ_j^t, the number of training samples x_j^t and the number of local epochs E_j. Thus they can be calculated as following:

$$\begin{cases} \tau_{j,com}^t = \frac{\phi}{r_{j,d}^t} + \frac{\phi}{r_{j,u}^t} \\ \tau_{j,c}^t = \frac{1}{\rho_j^t \cdot f_j^t} \cdot E_j \cdot x_j^t \end{cases}. \tag{5}$$

The maximum waiting time τ_s^t for each round t is determined by the latest submitted vehicular node. Hence, the server waiting time at global round t is calculated by:

$$\tau_s^t = \max_{j \in S^t} \left(\tau_{j,c}^t + \tau_{j,com}^t \right), \tag{6}$$

where S^t refers to the participating nodes set at t round.

In FedAvg protocol, server randomly selects a certain proportion of uploaded local models to perform averaging aggregation. Therefore, the resource consumption of the whole FL process can be expressed by:

$$R = \sum_{t=1}^{T} \sum_{j=1}^{N_{ab}^t} \left(f_j^t \cdot \tau_{j,c}^t + \tau_{j,com}^t \right), \tag{7}$$

where N_{ab}^t is the abandon node set at global round t that execute training tasks locally without participating in global aggregation.

Local loss function: For each selected vehicular node, let $\ell(w_j)$ denote the loss functions of local model j under deep neural network with local dataset D_j which can be calculated by:

$$\ell(w_j) = \frac{1}{n_j} \sum_{x_i, y_i \in D_j} \ell\left(w_j; x_i, y_i\right), \tag{8}$$

where n_j denotes the sample number of node j.

These vehicular nodes independently update the local loss with executing multiple steps of stochastic gradient descent (SGD) on local dataset.

Global loss function: For the parameter server, according to the training process of FL, global model loss $\mathcal{L}(w)$ can be expressed as following:

$$\mathcal{L}(w) = \sum_{j=1}^{M} \frac{n_j}{n} \ell(w_j), \tag{9}$$

where n is the total sample number of all nodes and satisfies $n = \sum_{j=1}^{M} n_j$. Therefore, the objective function of the FL-based IoV system with constraints can be expressed as

$$\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{j=1}^{M} \sum_{x_i, y_i \in D_j} \ell\left(w; x_i, y_i\right),$$

$$\text{s.t.} \sum_{t=1}^{T} \max_{j \in S^t} \left(\tau_{j,c}^t + \tau_{j,com}^t \right) \le T_{\max}$$

$$\sum_{t=1}^{T} \sum_{j=1}^{N_{ab}^t} \left(f_j^t \cdot \tau_{j,c}^t + \tau_{j,com}^t \right) \le R_{\max},$$

where $T_{\max}$ and $R_{\max}$ are constants that denote the maximum acceptable global training time and resource waste of FL respectively.

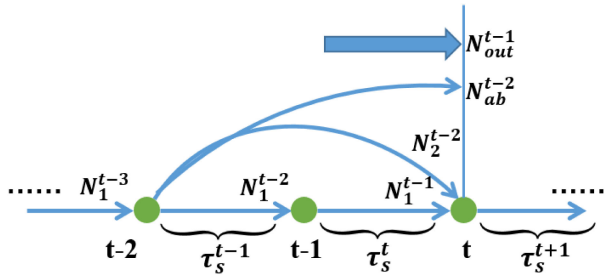


Fig. 2. Illustration of vehicular nodes communication.

It is difficult to find a closed-form solution for the objective function due to the complicated inherent characteristic of many machine learning model. Therefore, we exploit to design some novel solutions in Section IV to reduce training time and resource cost of federated learning while maintaining learning accuracy.

IV. Methodology

To solve the aforementioned problems, in this section, the Semi-SynFed protocol is proposed with two folds, namely client selection and dynamic aggregation.

A. Client Selection

The client selection scheme is proposed to ensure that the ICV acted as a federated learning node can have the stable performance and network status. Meanwhile, it can guarantee that the participating vehicles have valuable samples, and reduce the waste of computation resources by terminating the learning clients whose data are no longer important for the model training. In FL procedure, there is always a certain number of vehicular nodes, which cannot submit parameters within the required time in each global round, due to the delay in a poor communication environment or shutdown of the vehicle. To show the latency of nodes in different ranges, we denote several node sets with different latency levels as shown in Fig. 2. We denote τ_s^t as the maximum server waiting time at global round t, and N_1^{t-1} represents a set of vehicle nodes which have completed uploading task within one global round after the global round $(t-1)$. N_2^{t-2} are used to represent the vehicle nodes which have completed uploading task by more than one round but within two rounds after the global round $(t-2)$.

Furthermore, considering the global round t, N_{ab}^{t-2} denotes a set of the vehicle nodes which cannot completed uploading task over two rounds from global round $t-2$, and N_{out}^{t-1} is the node set excluded by client selection at the $t-1$ round.

Then let O^t represent a set of nodes which can respond to the learning task at global round t, and it can be represented as

$$O^t = N_1^{t-1} + N_2^{t-2} + N_{ab}^{t-2} + N_{out}^{t-1}. \tag{10}$$

In Semi-SynFed, we further consider computational capability (CC) and network capability (NC) to evaluate the state of ICV dynamically. CC is an indicator to measure the overall status of device performance and computing resources. NC is an indicator to measure the communication speed. In general, the transmission power of ICV is less than BS, which makes the uploading rate of ICV become the bottleneck of communication. Therefore, the instantaneous upload rate of vehicle is selected to characterize NC. By setting threshold, vehicles with low CC and high NC are selected to join in FL procedure. The formula of CC and NC is as following:

$$\begin{cases} CC_j^t = \frac{1}{\rho_j^t \cdot f_j^t} \\ NC_j^t = r_{j,u}^t = B \cdot \log_2\left(1 + \frac{\mathrm{P}_{veh}}{N \cdot d_t^2 \cdot \eta_0}\right) \end{cases}. \tag{11}$$

Furthermore, to select the nodes who have bigger contribution to the global model, we consider gradient norm of local model at each round. For the Convolutional Neural Networks (CNNs) training, gradient norm can measure the improvement of training data to model parameters. However the calculation of gradient norm for all parameters are extremely complications. To reduce the computation complexity of back propagation, we choose the gradient norm σ_j of the last layer, which has been shown in the recent research [30] that the last layer gradient can represent the importance level of the local data. The gradient norm is calculated as

$$\sigma_j = \gamma \left\| \frac{\partial \ell(w_{last,j})}{\partial w_{last,j}} \right\|_2^2, \tag{12}$$

where $w_{last,j}$ is the weight of last layer in the node j.

The gradient norm of the penultimate layer can be used as the measurement since the penultimate layer includes the highest features of the input.

The details of client selection are described in Algorithm 1. At first,the input the algorithm is several parameters, including the node set O^t, the maximum waiting time τ_s^{t+1}, model size ϕ, and so on. The output is the selected node set S^t. For each node c_j^t in O^t, the computing capability CC_j^t, network capability NC_j^t and gradient norm σ_j^t are calculated at client side according to the state parameters device performance ρ_j^t, computation resource f_j^t and uploading rate $r_{j,u}^t$ (line 3-8 of Algorithm 1). After receiving CC_j^t, NC_j^t and σ_j^t of node j, the server determines whether the node is selected into S^t in line 9-13 of Algorithm 1.

Algorithm 1: Client Selection.

Input: Response node set O^t, maximum server waiting time τ_s^{t+1}, training model size ϕ, threshold of computing capability and gradient norm CC_{max}, σ_{max}.
Output: The selected set at S^t

1: set $S^t = \varnothing$
2: **for** each client $c_j^t \in O^t, j = 1, 2, \ldots$ **do**
3: **// Client Side:**
4: Get the current state parameter ρ_j^t, f_j^t and $r_{j,u}^t$
5: Computing capability: $CC_j^t = \frac{1}{\rho_j^t \cdot f_j^t}$
6: Network capability: $NC_j^t = r_{j,u}^t$
7: Gradient norm: $\sigma_j^t = \gamma \| \frac{\partial \ell_j(w_{last,j})}{\partial w_{last,j}} \|_2^2$
8: Upload CC_j^t, NC_j^t and σ_j^t to server.
9: **// Server Side:**
10: Receive CC_j^t, NC_j^t and σ_j^t
11: **if** $CC_j^t < CC_{max}$ and $NC_j^t > \frac{\phi}{2 \cdot \tau_s^t}$ and $\sigma_j^t > \sigma_{max}$ **then**
12: $S_t = S_t \cup \{c_j^t\}$
13: **end if**
14: **end for**

B. Dynamic Aggregation

1) Aggregation Scheme: Generally, the calculation process of FL takes place on the server side and the client side, respectively. On the client side, ICVs could download the global model parameters w^{t-1} from a parameter server, and update it by stochastic gradient descent (SGD) which is express as:

$$w_j^t = w^{t-1} - \eta \sum_{i=1}^{D_j} \nabla \ell\left(w^{t-1}; x_i, y_i\right), \tag{13}$$

where η is learning rate that is used to control how much loss to change at each iteration and w_j^t is local model parameter of node j.

Considering the dynamic network and heterogeneous computation capacity, in this paper, we design an asynchronous aggregation method based on gradient aggregation to balance the trade off between time consumption and computation resources. In Semi-SynFed, new global parameters w^{t_g+1} can be calculated as the sum of the last round of global parameters w^{t_g}. Let $\alpha_j^{t_j}$ denotes to the mixing hyper-parameter, which is equal to the multiplication of the proportion of training samples and the linear function of staleness. Therefore, we can get the following updated global weight by:

$$w^{t_g+1} = w^{t_g} + \sum_{j \in N^{t_g}} \alpha_j^{t_g} \cdot \nabla w_j^{t_j}, \tag{14}$$

where t_g and t_j denote server round and local node round, respectively. $\nabla w_j^{t_j}$ is the gradient of node j at t_j round. N^{t_g} is the node set that are successfully uploaded in t_g round and satisfied $t_g - t_j \leq 1$.

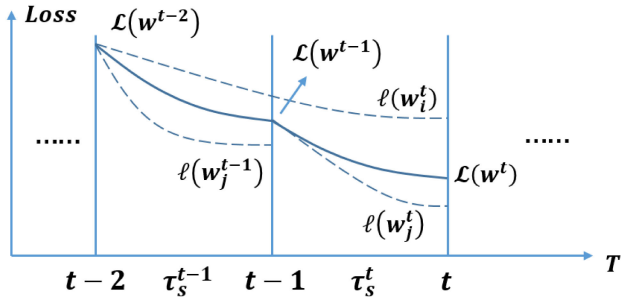


Fig. 3. Local loss and global loss at each round.

Furthermore, $\alpha_j^{t_g}$ denotes the weight of gradient of node j at t_g round. $\alpha_j^{t_g}$ can be calculated by:

$$\alpha_j^{t_g} = \frac{n_j}{n^{t_g}} \cdot \frac{1}{a \cdot (t_g - t_j) + 1}, \tag{15}$$

where n_j is the number of samples of node j. n^{t_g} is the total sample number of all participating nodes at global round t_g and satisfies $n^{t_g} = \sum_{j \in N^{t_g}} n_j$. a is staleness decay.

When all selected vehicles reach the maximum server waiting time, the rounds of vehicles are the same as the rounds of server ($t_g = t_j$). According to (14) and (15), the aggregation can be summarized as:

$$w^{t_g+1} = w^{t_g} + \sum_{j \in S^{t_j}} \frac{n_j}{n^{t_j}} \cdot \nabla w_j^{t_j}. \tag{16}$$

In this context, the aggregation scheme is similar to FedAvg whose aggregation is $w^{t+1} = \sum_{j \in S^t} \frac{n_j}{n^t} \cdot w_j^{t+1}$ [13]. It means that the accuracy and communication cost of proposed scheme will be similar to FedAvg under good communication conditions. Otherwise, server aggregates the parameters of vehicular nodes uploading in different global rounds according to the value of $t_g - t_j$. Hence, the parameter server can aggregate the local models of slow vehicular nodes asynchronously, which is conducive to the distribution diversity of training samples.

2) Convergence Analysis: As shown in Fig. 3, the dash line represents local loss, and the solid line represents the change of global loss. Let $w_i^t \in N_2^{t-2}$ and $w_j^t \in N_1^{t-1}$. w_i^t is the updated model obtained by w^{t-2} through B_i' rounds of local iteration. w_j^t is the updated model obtained by w^{t-1} through B_j' rounds of local iteration. In global round t, the server aggregates can get w^t through (14). Assuming that the global loss $\mathcal{L}(w)$ is L-smooth and μ-strong convex. Define $G(T) = \mathcal{L}(w^T) - \mathcal{L}(w^*)$, after T global rounds, the proposed method can converge to the global optimum w^* as following:

$$G(T) \leq d_1 G(T-1) + s^{T-1}\left[G(1) - d_1 G(0) + d_2\right] - d_2, \tag{17}$$

where $d_1 = \frac{-2\lambda}{\sqrt{a^2+4\lambda(a+1)}+a}$, $s = \frac{2\lambda}{\sqrt{a^2+4\lambda(a+1)}-a}$, $d_2 = \frac{L^2\eta^2\left(\sqrt{a^2+4\lambda(a+1)}+a+2\right)}{4(\lambda-1)(L-\mu)}\|\nabla\ell(w^0) + \frac{1}{L\eta}\nabla\mathcal{L}(w^0)\|^2$, $\lambda = (1 - \frac{\mu}{L})^{\mathcal{B}}$. $\mathcal{B}$ is the rounds of local iteration, η is learning rate, and a is staleness decay. The detailed proof is shown in Appendix A.

3) Maximum Server Waiting Time: At each global round, the parameter server will wait for all nodes to upload parameters before performing aggregation. When the transmission rate of some nodes is low, the iteration speed of the global model will decrease. Hence, the maximum server waiting time is usually set to reduce the impact of high-latency nodes. However, the fixed waiting time designed in advance may not be the most suitable for the current communication network. Especially when the condition of the communication network changes frequently, the fixed waiting time may make the FL perform worse.

To balance the overall change of communication network and the impact of high-latency nodes, we design a dynamically adjusted server waiting time. In each global round, the server aggregates the local parameters that staleness does not exceed two rounds. But we hope that enough clients can complete training and upload within one round. We define the proportion of clients uploaded in one round at global round t is:

$$ar^t = \frac{\left|N_1^{t-1}\right|}{\left|S^{t-1}\right|}. \tag{18}$$

In the FL procedure, we hope that the proportion ar^e of aggregate nodes at each round is stable. Hence, the expected server waiting time is

$$\tau_s^e = \frac{\phi}{B\log_2\left(1 + \frac{n_a \pi P}{S_a N \eta_0 \cdot ar^e}\right)} + \tau_{j,c}^t, \tag{19}$$

where $\tau_{j,c}^t$ is average local computation time that is expressed as:

$$\tau_{j,c}^t = \tau_{j,a}^t \frac{E_j \cdot x_j}{\rho_j \cdot f_j^t}\beta_3, \tag{20}$$

where $\tau_{j,a}^t$ is actual local training time. ρ_j and f_j^t are set to a random number but changing with global round. β_3 is adjustment factor.

However, ar^t is unstable due to the influence of some uncontrollable factors such as road restrictions and local computation time. Considering the continuity of vehicle movement, we exploit ar^t to predict the expected waiting time τ_s^{t+1}. So we have

$$\tau_s^{t+1} = \tau_s^t + \Delta\tau^t(ar^t) = \tau_s^t + \frac{\phi}{B}\left(\frac{1}{f(ar^t)} - \frac{1}{f(ar^e)}\right), \tag{21}$$

where $f(x) = \log_2(1 + \frac{n_a \pi P}{S_a N \eta_0 \cdot x})$. To simplify calculation, let $\Delta\tau^t(ar^t) \approx k(ar^e - ar^t)$, where k is a constant related to simulation setting. Therefore, we assume that $ar^e - ar^t$ follows logistic distribution to limit the maximum and change speed of server waiting time. So we have

$$\tau_s^{t+1} = \tau_s^t + \beta_2 \cdot \frac{e^{\beta_1 \cdot (ar^e - ar^t)} - 1}{e^{\beta_1 \cdot (ar^e - ar^t)} + 1}, \tag{22}$$

where β_1 and β_2 are relative coefficients. The detailed derivations of (19), (21) and (22) are shown at Appendix B.

Through client selection scheme and the dynamic waiting time scheme, the parameter server can pre-identify the nodes which is good for the learning task, and adjust the maximum waiting time according to overall changes of communication network, to achieve more efficient FL procedure.

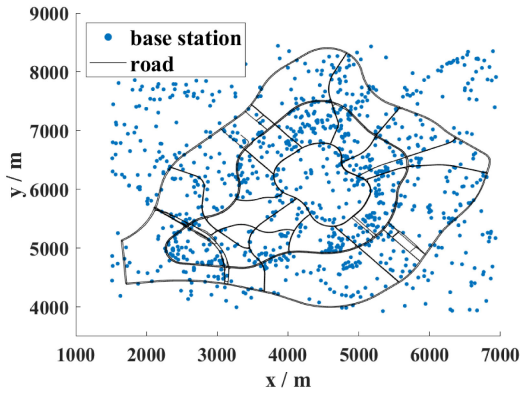


Fig. 4. The simulation map of road network and BS position.

The details of the Semi-SynFed protocol are shown in Algorithm 2. At first, the server builds the initial model w^0 according to machine training task. O^0 is the node set initially participating in FL procedure. The server broadcasts model w^0 to the nodes in O^0 (line 2-3). FL procedure can be divided into server side and client side. At server side, the server aggregates the uploaded models $w_j^{t_j}$ from node set N^t after waiting τ_s^t at each communication round t (line 5-11). The waiting time τ_s^{t+1} and the selected node set S^t for the next round are calculated at line 18-22. And then the server broadcasts the new global model w^t to S^t (line 23). At client side, each vehicular node at S^t performs SGD to update local models $w_j^{t_j}$ (line 28-34) at each communication round. The testing loss $\ell(w_j)$ is calculated (line 35), and then the updated local model $w_j^{t_j}$ and the testing loss $\ell(w_j)$ are uploaded to the server. There are two end conditions for FL procedure, one is when global model loss $\mathcal{L}(w^t)$ is lower than the expected loss L^e (line 13-17), and the other is when the running rounds exceeds the maximum number of rounds (line 25-26).

Algorithm 2: Semi-SynFed Protocol.

Input: Initial node set O^0, initial model w^0, initial maximum server waiting time τ_s^1, learning rate η, batch size B, model size ϕ, relative coefficient β_1 and β_2, expected global loss L^e
Output: Global optimal model w^*

1: **Server Side:**
2: Set $S^0 = O^0$
3: Broadcasting w^0 to S^0.
4: **for** each global round $t = 1, 2, \ldots, T$ **do**
5: The node set of uploading model in time τ_s^t:
6: $N^t = N_1^{t-1} \cup N_2^{t-2}$
7: $n^t = \sum_{j \in N^t} n_j$
8: **for** client $c_j^t \in N^t, j = 1, 2, \ldots$ **do**
9: $\alpha_j^t = \frac{n_j}{n^t} \cdot \frac{1}{a \cdot (t - t_j) + 1}$
10: $w^t = w^{t-1} + \alpha_{t_j} \cdot \nabla w_j^{t_j}$
11: $t_j = t$
12: **end for**
13: Global model loss: $\mathcal{L}(w^t) = \sum_{j=1}^{M} \frac{n_j}{n^t} \ell(w_j)$
14: **if** $\mathcal{L}(w^t) < L^e$ **then**
15: $w^* = w^t$
16: **return** w^*
17: **end if**
18: Maximum server waiting time:
19: $ar^t = \frac{|N_1^{t-1}|}{|S^{t-1}|}$
20: $\tau_s^{t+1} = \tau_s^t + \beta_2 \cdot \frac{e^{\beta_1 \cdot (ar^e - ar^t)} - 1}{e^{\beta_1 \cdot (ar^e - ar^t)} + 1}$
21: Get the node set that can respond to server O^t.
22: $S^t =$ ClientSelection $(O^t, \tau_s^{t+1}, \phi)$
23: Broadcasting global model w^t to S^t.
24: **end for**
25: $w^* = w^T$
26: **return** w^*
27: **Client Side:**
28: **for** each client $c_j^t \in S^t, j = 1, 2, \ldots$ **do**
29: Downloading global model w^t.
30: Initializing $t_j = t, w_j^{t_j} = w^t$
31: $n_j = \|D_j\|, n_j^{test} = \|D_j^{test}\|$
32: **for** sample x_i, y_i in D_j^{train} **do**
33: $w_j^{t_j} = w_j^{t_j} - \eta \nabla \ell(w_j^{t_j}; x_i, y_i)$
34: **end for**
35: Test loss: $\ell(w_j) = \frac{1}{n_j^{test}} \sum_{x_i, y_i \in D_j^{test}} \ell(w_j^{t_j}; x_i, y_i)$
36: Uploading $w_j^{t_j}$, t_j, n_j and $\ell(w_j)$ to server.
37: **end for**

V. Experiment

In this section, we first provide the details of simulation settings of IoV and training tasks, including parameter settings and data redistribution. Sequentially, we investigate the accuracy, convergence speed, and resource consumption of different protocols in case of different training tasks.

A. Simulation Setting of IoV

To build a simulation environment of IoV close to the real world, we use real road network data to build a simulated road network. We obtain the real road network data from open street map database [31] and the location data of wireless communication base station (BS) from openGPS database [32]. The area of both two data is Guangzhou Higher Education Mega Center, a university town in China. The scale of the selected area is $5.5km * 4.5\ km$ and the number of BS is 1396. By transforming the latitude and longitude coordinates of BS into road network coordinates, we have a simulation map integrating road network and BS, which is depicted in Fig. 4. For the sake of privacy issues, a small random deviation is added to the location of BS, which does not affect the experiment results. To facilitate the data acquisition, we assume that the communication of IoV is realized through the wireless communication base stations that work as roadside communication devices.

We apply the Simulation of Urban MObility (SUMO) [33], an open-source traffic simulation platform, to perform the vehicle motion in the simulated road network. According to the industry standard of IoV the bandwidth is set 20 MHz, and the transmitting power of roadside devices and ICV is set 29 dBm and 26 dBm, respectively. The number of ICV is 944. Running

TABLE II
DETAILS OF DATASETS AND MODELS

Parameter	Task 1	Task 2	Task 3
Dataset	FMNIST	cifar10	TSRD
Dataset size	70,000	60,000	6,164
Labels	10	10	58
Image size	28*28*1	32*32*3	96*96*3
Model size	6.38 MB	36.88 MB	22.46 MB

time is 30,000s. Spatial noise and propagation factor are set 0.002 w and 1.

The transmission rate of vehicular nodes at each time slot can be calculated according to (3). Through a testing, the average transmission rate of downloading and uploading are 6.41 MB/s and 3.54 MB/s, respectively, which is close to the real LTE-Advanced [34]. For the model training, we use Keras to build neural models under tensorflow framework and NVIDIA RTX 2080Ti is used to accelerate the training process.

B. Training Tasks and Data Redistribution

In order to verify the efficiency of the Semi-SynFed protocol, we consider three types of machine learning tasks. All the training tasks are related to image classification, which are based on three different datasets (Traffic Sign Recognition Database (TSRD), Fashion-MNIST [35] (FMNIST), and Cifar10).

Different from traditional distributed machine learning, the federated learning protocol has to consider the heterogeneity of data distribution among nodes. Therefore, it's necessary to form a Non-Independent and Identically Distributed (Non-IID) dataset among nodes. To achieve data redistribution, all samples are randomly divided into a certain number of subsets according to the dirichlet distribution.

The samples in each subset are unique and non-repetitive. The label distribution in each subset and the total number of samples between the subsets are Non-IID. Then, selecting a part of subsets randomly and the total sample number of these subsets is bigger than n_j. These selected subsets constitute the local dataset of client j.

To compare the impact of different scale training tasks on the FL protocols, we benchmark over the model size by different convolutional neural networks and the total sample number of different datasets. The hyperparameters of CNN such as learning rate, delay, and batch size are set to $1e-2$, $1e-6$, and 20, respectively. The details of datasets and models are shown in Table II.

C. Performance in Different Training Tasks

In this subsection, we mainly analyzes the results of the Semi-SynFed protocol and other three FL protocols at three training tasks in terms of convergence speed, communication efficiency and resource consumption.

We compare the Semi-SynFed protocol with other three FL protocols that are FedAvg [13], FedAsync [23], and SAFA [16]. In Semi-SynFed, staleness decay a and expected aggregation ratio ar^e are set to 0.3 and 0.8. Other hyperparameters β_1, β_2, CC_{max}, and σ_{max} are set to 5, 30, 80, and 0.1, respectively. In FedAvg, the aggregation rate is set to 0.9. In FedAsync, a linear staleness function is applied when the two parameters of staleness function α and a are set to 0.8 and 2. Max staleness and maximum server waiting time are set to 5 and 100 s. SAFA can be considered as an improved asynchronous method. In this experiment, the original client selection of SAFA is removed for simplicity and the maximum server waiting time is set to 100 s. The local training epoch of all protocols is set to 1.

Convergence speed : As shown in Fig. 5, for all learning tasks, the Semi-SynFed can make global model converge stably and achieve the same accuracy as FedAvg but faster than FedAvg. In addition, as shown in Fig. 5(c), the convergence speed of Semi-SynFed is behind FedAsync and SAFA. The possible reason is a larger model required in TSRD. Fig. 6 represents the test accuracy of global model with various types of runtime. As shown in Fig. 6, the Semi-SynFed can achieve the expected accuracy within the maximum runtime, which is similar to the change of test loss. The convergence speed of Semi-SynFed in small model training is faster than that in larger models.

Communication efficiency: As shown in Fig. 7, in all training tasks, the communication rounds of Semi-SynFed are much smaller than FedAsync, and in the same order of magnitude as FedAvg and SAFA. It shows that Semi-SynFed can achieve the same accuracy with less communication rounds when comparing with asynchronous method.

Resource consumption : We further evaluate the resource consumption situation. As shown at Fig. 8, the resource consumption of Semi-SynFed is lower than FedAsync and FedAvg apart from cifar10. Compared with SAFA, the Semi-SynFed performs better in large model training. When it comes to the small model training of FMNIST, the resource consumption of SAFA basically has no change during the entire distributed training process since the maximum server waiting time for initialization meets the requirements of training tasks.

As a summary, the Semi-SynFed can converge in three distributed training tasks with the Non-IID setting. For convergence speed, compared to FedAvg in case of the training task TSRD, the Semi-SynFed is 20% faster while the resource consumption is almost same. The performance of Semi-SynFed in other training tasks is also better than FedAvg. Although the Semi-SynFed is slower in convergence speed when comparing to asynchronous methods (i.e. FedAsync, SAFA), there is a significant dropping in resource consumption, which means that the Semi-SynFed can achieve good performance on the balance of convergence speed, accuracy and resource consumption.

In the IoV communication environment, the resource consumption of the Semi-SynFed with dynamic maximum waiting time performs worse than SAFA with the fixed maximum waiting time. The reason is that the expected aggregation ratio is too small. The expected aggregation ratio set by Semi-SynFed represents the a certain ratio of nodes whose uploading time is larger than one global round time. When the expected aggregation ratio is small, there will always be some nodes that take more than two global rounds to respond to the server. In small-scale model training, the local training time is smaller than the communication time. Thus each node can finish local training task faster, which means that more nodes can

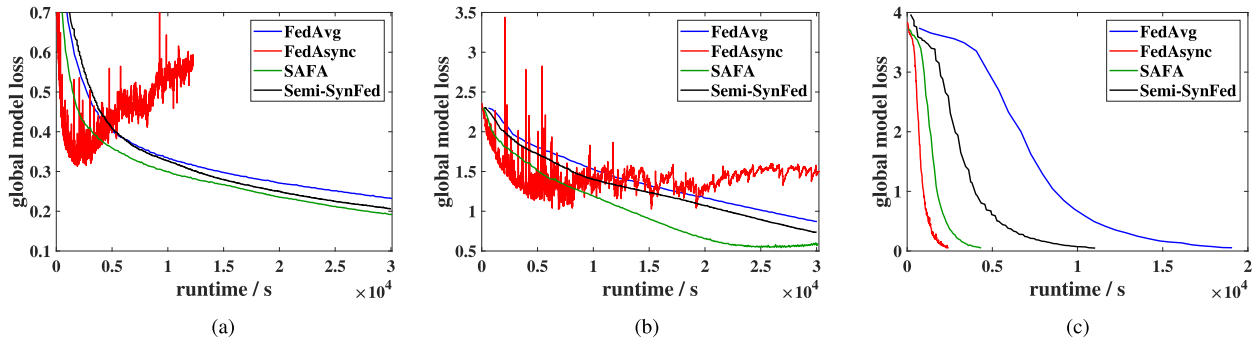


Fig. 5. Global model loss with runtime for different training tasks. (a) FMNIST. (b) cifar10. (c) TSRD.

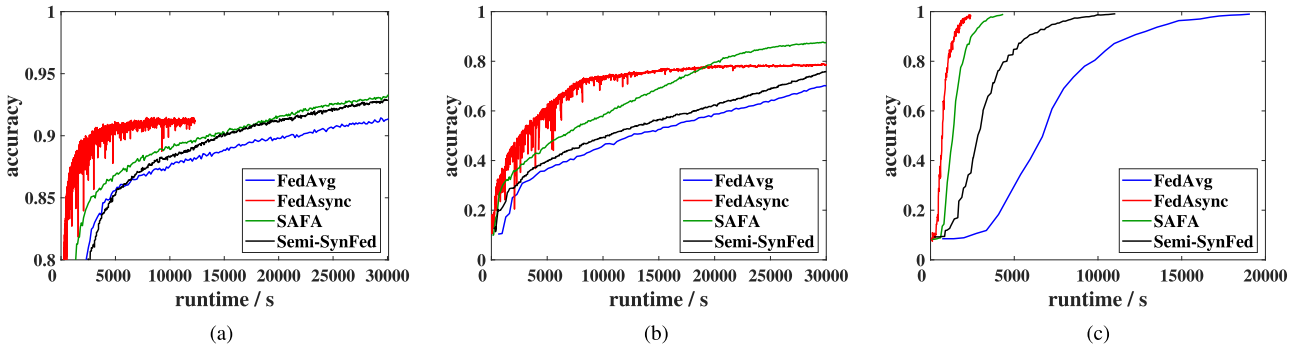


Fig. 6. Global model accuracy with runtime for different training tasks. (a) FMNIST. (b) cifar10. (c) TSRD.

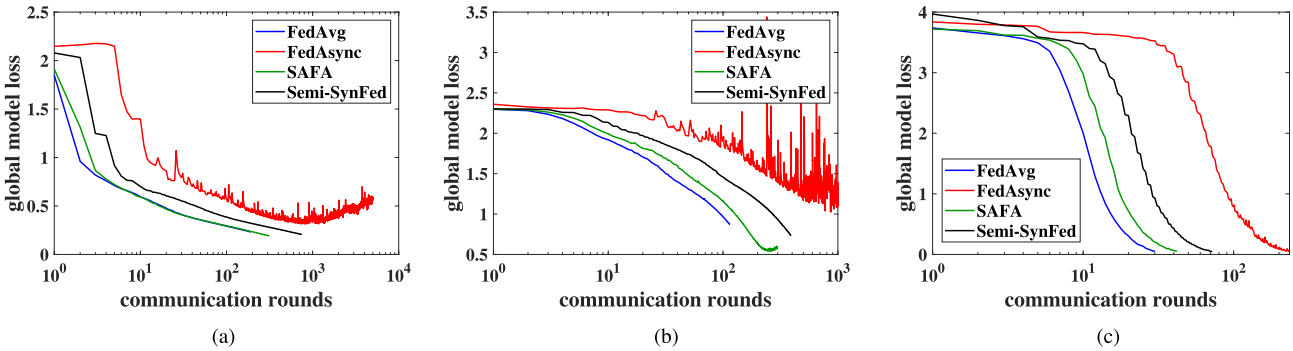


Fig. 7. Global model loss with communication rounds for different training tasks. (a) FMNIST. (b) cifar10. (c) TSRD.

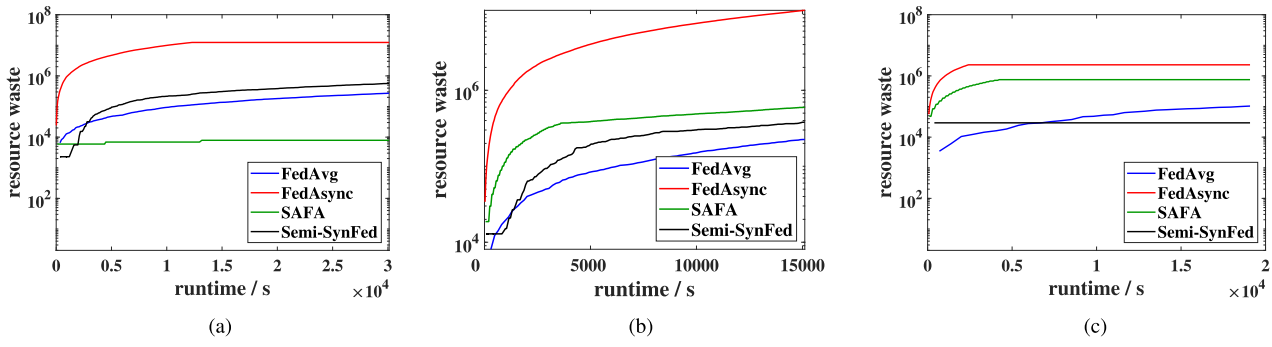


Fig. 8. Resource consumption with runtime for different training tasks. (a) FMNIST. (b) cifar10. (c) TSRD.

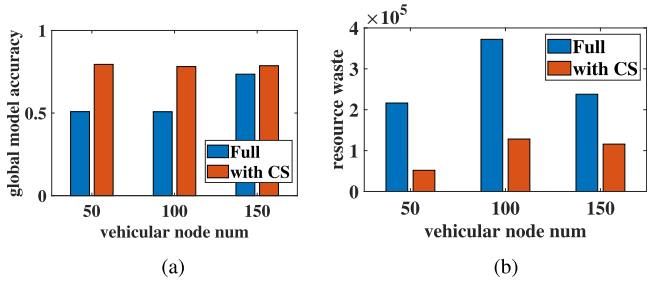


Fig. 9. The influence of client selection with different number of nodes. (a) Global model accuracy. (b) Resource consumption.

participate in global aggregation under the same communication environment.

D. Influence of Client Selection

To evaluate the effectiveness of the proposed Client Selection (CS) scheme, we consider the Semi-SynFed with CS and the Semi-SynFed with full broadcast (broadcast to all nodes that respond to the server at every global round) over three different numbers of nodes. We measure CS at task 3 with TSRD and the runtime is set to 3,600 s.

Regarding on the model accuracy, as shown in Fig. 9(a), Semi-SynFed with CS is higher than Semi-SynFed with full broadcast. The main reason is that the application of CS reduces the nodes with poor performance, which improves convergence speed. In terms of resource consumption, CS can significantly reduce the resource consumption in different numbers of nodes, as shown in Fig. 9(b).

To further verify the effectiveness of the proposed CS scheme, the number of response node set O^t and selected node set S^t are shown in Fig. 10(a) the number of O^t is basically maintained at a constant value, which means that a certain percentage of clients always have a stable communication rate in IoV. With the communication round increasing, the number of S^t gradually decreases. Due to the global model gradually converges, more and more nodes don't satisfy the gradient norm condition ($\sigma_j^t > \sigma_{max}$). This results show that the client selection scheme works well to remove the nodes without valuable data timely to improve training speed and reduce resource consumption.

To further verify the impact of gradient norm, We evaluated the model accuracy and resource consumption with or without gradient norm. As shown in Fig. 11(a), with a bigger number of clients, such as the number of nodes is set to 100 and 150, gradient norm can significantly improve the convergence speed. Regarding on the resource consumption as shown in Fig. 11(b), gradient norm condition also can reduce it clearly for different amount of nodes.

E. Influence of Dynamic Maximum Waiting Time

To verify the effectiveness of the dynamic maximum waiting time scheme, we have performed the evalaution by varying the maximum waiting time and the number of each node set under Semi-SynFed protocol with the proposed CS scheme. In Fig. 10(b), we can see that the number of N_1^{t-1} is close to the number of S^t in all global rounds, and we can see that the maximum waiting time can ensure that a certain percentage of nodes from S^t can complete uploading within one round at each global round in Fig. 10(c). It means that the dynamic maximum waiting time can promise that there are enough nodes to participate in the global aggregation at dynamic communication network.

As shown in Fig. 10(b) and (c), the dynamic maximum waiting time and client selection scheme optimize the FL process from different aspects. The dynamic maximum waiting time scheme is concerned with the training time and upload time of the nodes from the selected node set S^t. The CS scheme is concerned with device and network performance and data learning value of the nodes from response node set O^t. Through the effects of the two schemes, the nodes with poor performance and low learning value are continuously removed from the federated training network, which means the impact of bad nodes is reduced. The maximum waiting time of each round is decreased and the number of unresponsive nodes for more than two rounds is reduced, thereby achieving the goal of improving the convergence speed and saving resources.

TABLE III
THE INFLUENCE OF NODE NUMBER

node num	FL protocol	Accuracy(%)	Resource consumption ($\times 10^4$)
50	FedAvg	18.3	1.6
	FedAsync	98.8	230.3
	SAFA	89.3	7.5
	Semi-SynFed	79.4	5.1
100	FedAvg	23.6	6.5
	FedAsync	99.1	531.2
	SAFA	88.8	26.2
	Semi-SynFed	78.1	12.8
150	FedAvg	28.0	9.3
	FedAsync	99.1	583.0
	SAFA	86.0	23.2
	Semi-SynFed	78.6	11.5

F. Number of Vehicular Nodes

In the IoV system, as the number of vehicular nodes increases, the total number of training samples increases, which means that the learning model can learn more knowledge from complex traffic scenarios. We evaluate the number of vehicular nodes in three degrees of 50, 100, and 150 in training tasks with TSRD. We compare global model accuracy and resource consumption of four FL protocols when the runtime is 3,600 s. The results are shown in Table III.

We can see that the accuracy of Semi-SynFed is stable among the three different numbers of nodes and is about 10% lower than SAFA's. For the resource consumption, the Semi-SynFed's is lower than the other two asynchronous methods. When the number of nodes are 100 and 150, the resource consumption of Semi-SynFed is only half of SAFA. It can be found that as the number of vehicular nodes increases, the effect of Semi-SynFed in saving resources of computation and communication becomes more and more significant.

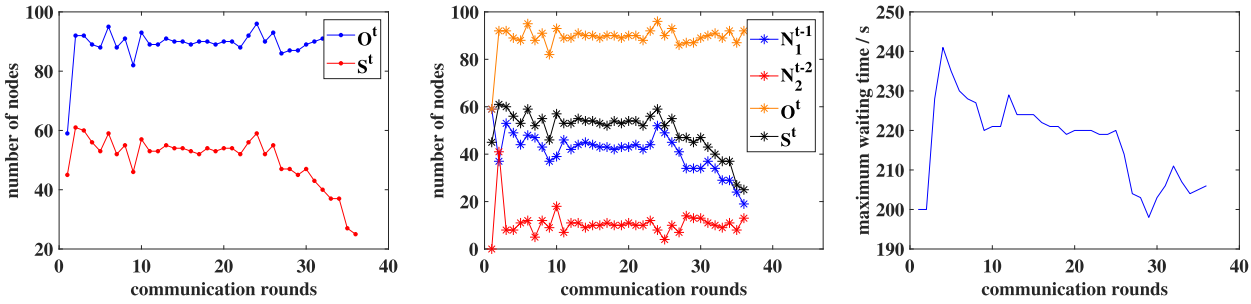


Fig. 10. Number of nodes in different node sets and the maximum server waiting time with communication round. (a) The number of nodes in O^t and S^t. (b) The number of nodes in N_1^{t-1}, N_2^{t-2}, O^t and S^t. (c) The maximum server waiting time.

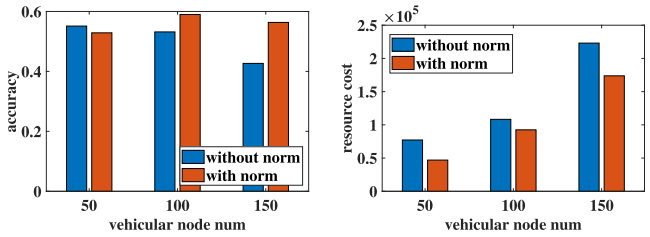


Fig. 11. Impact of gradient norm. The test accuracy and resource consumption of global model is selected when runtime is 3,600 s.

Fig. 12. Data preprocessing of front image. (a) Before. (b) After.

G. End-to-End Autonomous Driving

To further verify the advantage of the proposed method compared with central learning, we extend an end-to-end autonomous driving model with an open dataset apolloscape (v1.5), whose size is about 1,280,000 [36].

For the autonomous driving, the main challenge is to handle complex traffic scenarios (such as extreme weather and mixed traffic). Therefore, we apply federated learning to train autonomous driving model by collecting data within different traffic scenarios, which deals with various complex traffic scenarios.

In our experiments, a steering control model is built with front image (320×320×3) as input and curvature as output. The model aims to learn the operation of drivers in different traffic scenarios. We build a steering control model with the learning rate setting to 1e-3.

Before training, the original image is cropped up and downsampled to 66×200×3 in order to reduce the computing complexity. Fig. 12 shows the front image before and after preprocessing. Regarding to curvature, we assume the length of

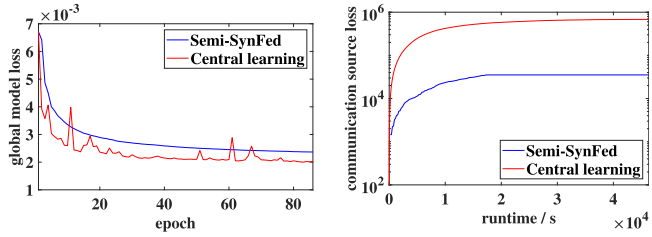


Fig. 13. Global model loss with epochs and communication consumption with runtime on apolloscape open dataset. (a) Global loss. (b) Communication consumption.

vehicles is 5 m and the maximum steering angle is 30°. Therefore, we eliminate the curvature beyond the range from -0.1 to 0.1. Hence, the curvature are magnified by 10 times to facilitate model training.

From Fig. 13(a), we can see that the final test loss of Semi-SynFed is close to that of central learning, which shows the feasibility of FL in IoV system. From Fig. 13(b), federated learning can greatly reduce communication consumption, while central learning requires all nodes to upload data to server, which represents the advantage by adopting FL in IoV system. The terminal time of federated learning and central learning is 17,269 s and 46,310 s respectively, which also proves that federated learning is more efficient.

H. Ablation Study

1) Bandwidth: To further verify the relationship between transmission bandwidth and global model accuracy, we evaluate different bandwidths values for federated learning training. As depicted in Fig. 14, we can find that the system runtime and resource consumption increase with the bandwidth decreases, which means that low bandwidth can reduce the speed of model convergence.

2) Expected Loss: L^e is one of the important terminating condition for federated learning. L^e represents the expected performance the global model can reach. To explore the impact of L^e, we apply different L^e in the same training task based on TSRD dataset, and the results are depicted in Fig. 15(a). As the expected loss L^e increases, the system runtime decreases. Therefore, when the global model can reach the expected loss,

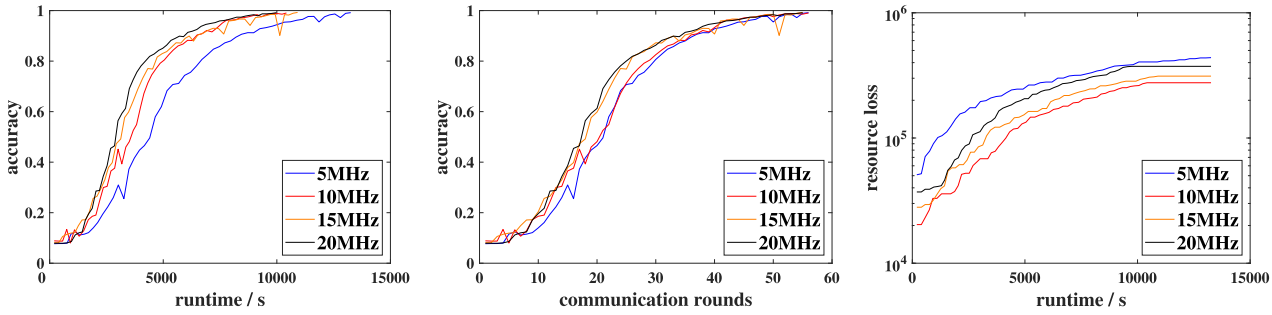


Fig. 14. Influence of different bandwidths.

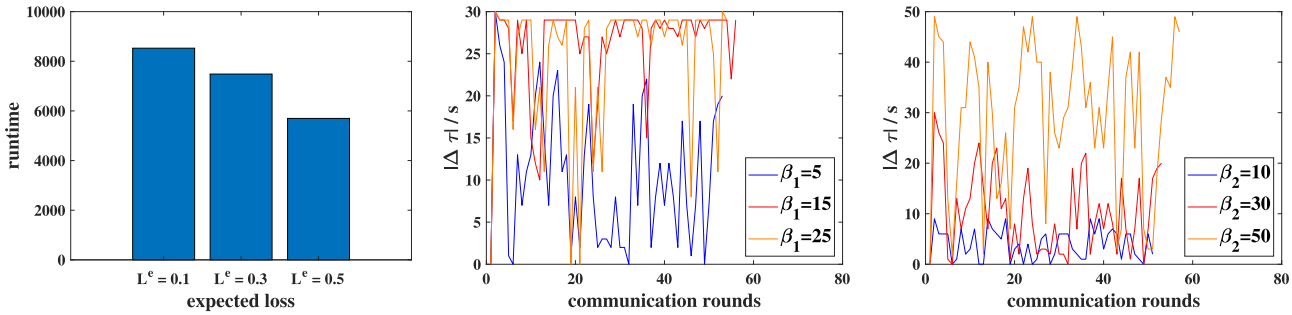
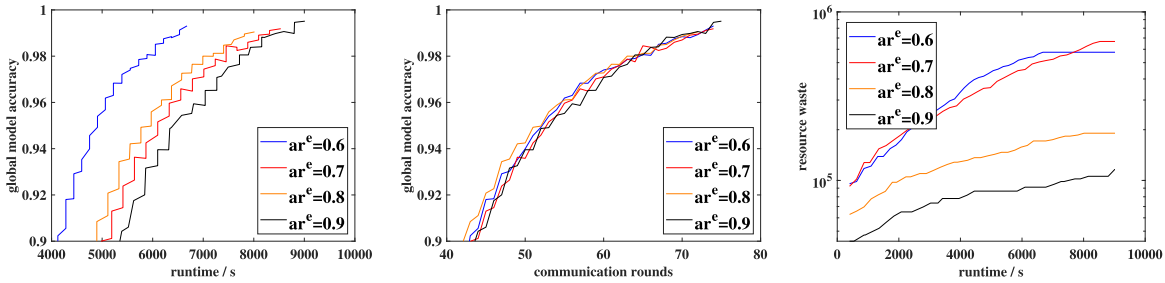


Fig. 15. Influence of L^e, β_1 and β_2 on TSRD dataset.



(a) (b) (c)

Fig. 16. Impact of different expected aggregation ratio ar^e. (a) Global model accuracy with runtime. (b) Global model accuracy with communication rounds. (c) Resource consumption with runtime.

the higher expected loss setting, the faster federated learning system can complete the training task.

3) Hyper Parameter β_1 and β_2: β_1 and β_2 are important parameters of calculation formula of maximum server waiting time τ_s^t. β_1 affects the changing sensitivity of $\Delta\tau^t(ar^t)$ and β_2 affects the Upper and lower limits of $\Delta\tau^t(ar^t)$. To explore the impact of β_1 and β_2 on $|\Delta\tau|$, we apply different β_1 and β_2 in the same training task based on TSRD dataset, and the results are depicted in Fig. 15(b) and (c). As shown in Fig. 15(b), as β_1 increases, $|\Delta\tau|$ reaches its maximum value more frequently. As shown in Fig. 15(c), as β_2 increases, the value range of $|\Delta\tau|$ becomes larger.

4) Expected Aggregation Ratio: The expected aggregation ratio ar^e is an important hyperparameter in Semi-SynFed that controls the number of nodes expected to complete uploading in one round by adjusting the maximum waiting time. The impact of ar^e is shown in Fig. 16. We can find that global model can converge faster with smaller ar^e. Meanwhile, in Fig. 16(b), we can see that there is almost no difference regarding on the number of communication rounds under different ar^e. The results show that the possible reason for reducing of converge time in Fig. 16 due to the reduction of maximum waiting time for smaller ar^e.

Furthermore, In Fig. 16(c), the resource consumption gradually decreases with the increasing of ar^e since bigger ar^e can ensure that most of nodes complete uploading within one round, so that few nodes need more than two rounds to complete uploading.

VI. Conclusion

In this paper, to solve the training problem by considering IoV environment, we designed a new Semi-SynFed protocol based on dynamic aggregation to improve the performance of federated learning. Through a unique client selection scheme,

Semi-SynFed can identify and remove bad nodes to improve communication efficiency. Meanwhile, by dynamically adjusting the maximum server waiting time, Semi-SynFed can ensure most nodes to complete uploading within two rounds, thereby reducing resource waste. In addition, Semi-SynFed updates global model through semi-synchronous gradient aggregation, which can improve the convergence speed. Finally, an IoV simulation environment is constructed with real road network data and mobile phone base station data to evaluate Semi-SynFed, by comparing with three typical FL protocols in terms of accuracy and resource consumption. Also the extensive experiments validate the effectiveness of client selection scheme and dynamic maximum waiting time scheme. For future work, we propose two main directions. (1) Explore the end-to-end data preprocessing of heterogeneous data and machine learning model with unlabeled data under FL framework, since it is an important research direction to realize FL applications. (2) More experiments of steering angle prediction will be done to verify the performance of FL protocols on different autonomous driving datasets.

APPENDIX A
CONVERGENCE ANALYSIS OF SEMI-SYNFED

We assume the global loss function $\mathcal{L}(w^t)$ is L-smooth and μ-strongly convex, so we have

$$\begin{cases} |f(y) - f(x) - \langle \nabla f(x), y - x\rangle| \le \frac{L}{2}\|y-x\|^2 \\ \frac{1}{2}\|\nabla f(x)\|^2 \ge \mu \cdot (f(x) - f(x_*)) \end{cases} \tag{23}$$

Since $\mathcal{L}(w^t)$ is convex, the global optimum exists. Let w^* denote the global optimum of $\mathcal{L}(w^t)$, i.e. $\mathcal{L}(w^*) = \min_w \mathcal{L}(w^t)$. At global round t, edge device j receives global model w^{t-1}. At each local iteration b in gradient descent algorithm, $w_{i,b}^{t-1}$ can be expressed as follows:

$$w_{i,b}^{t-1} = w_{j,b-1}^{t-1} - \eta \nabla \ell \left(w_{j,b-1}^{t-1} \right), \tag{24}$$

where η refers to learning rate.

Let $\mathcal{B}$ denote total rounds to do gradient descent in the client side. With the constraints of L-smooth and μ-strongly convex, for $\forall b \in \mathcal{B}$, the difference between local loss and optimal loss can be expressed as follows:

$$\begin{aligned} &\mathcal{L}(w_{j,b}^{t-1}) - \mathcal{L}(w^*) \\ &\le \mathcal{L}\left(w_{j,b-1}^{t-1}\right) - \mathcal{L}(w^*) + \eta \left\langle \nabla \mathcal{L}(w_{j,b-1}^{t-1}), -\nabla \ell(w_{j,b-1}^{t-1}) \right\rangle \\ &\quad + \frac{L\eta^2}{2} \left\| \nabla \ell(w_{j,b-1}^{t-1}) \right\|^2 \\ &\le \mathcal{L}(w_{j,b-1}^{t-1}) - \mathcal{L}(w^*) - \frac{1}{2L} \left\| \nabla \mathcal{L}(w_{j,b-1}^{t-1}) \right\|^2 \\ &\quad + \frac{L\eta^2}{2} \left\| \nabla \ell(w_{j,b-1}^{t-1}) + \frac{1}{L\eta} \nabla \mathcal{L}(w_{j,b-1}^{t-1}) \right\|^2 \\ &\le \left(1 - \frac{\mu}{L}\right) \left[\mathcal{L}\left(w_{j,b-1}^{t-1}\right) - \mathcal{L}(w^*) \right] + \frac{L\eta^2 C_1}{2}, \end{aligned} \tag{25}$$

where $\|\nabla \ell(w_{j,b-1}^{t-1}) + \frac{1}{L\eta} \nabla \mathcal{L}(w_{j,b-1}^{t-1})\|^2 \le C_1$. To simplify the description, let $G(t) = \mathcal{L}(w^t) - \mathcal{L}(w^*)$. After $\mathcal{B}$ rounds of local iteration, edge device j upload local model w_j^t to parameter server. So $\mathcal{L}(w_{j,0}^{t-1}) = w^{t-1}$ and $\mathcal{L}(w_{j,\mathcal{B}}^{t-1}) = w_j^t$. Hence, the difference between final local loss and optimal loss can be expressed as follows:

$$\begin{aligned} &\mathcal{L}(w_j^t) - \mathcal{L}(w^*) = \mathcal{L}(w_{j,\mathcal{B}}^{t-1}) - \mathcal{L}(w^*) \\ &\le \left(1 - \frac{\mu}{L}\right)^{\mathcal{B}} \left[\mathcal{L}(w_{j,0}^{t-1}) - \mathcal{L}(w^*) \right] + \frac{L\eta^2 C_1}{2} \sum_{b=1}^{\mathcal{B}} \left(1 - \frac{\mu}{L}\right)^{b-1} \\ &\le \left(1 - \frac{\mu}{L}\right)^{\mathcal{B}} \left[\mathcal{L}(w_{j,0}^{t-1}) - \mathcal{L}(w^*) \right] + \frac{L^2\eta^2 C_1}{2(L-\mu)} \\ &\le \left(1 - \frac{\mu}{L}\right)^{\mathcal{B}} G(t-1) + \frac{L^2\eta^2 C_1}{2(L-\mu)}. \end{aligned} \tag{26}$$

Let N_1^{t-1} denote the node set that are successfully uploaded in one round, and N_2^{t-2} denote the node set that are successfully uploaded over one round but no more than two rounds. Let N^t denote the node set that are successfully uploaded in t round and $N^t = N_1^{t-1} \cup N_2^{t-2}$. Therefore, with (15), the process of aggregation can be expressed as following:

$$\begin{aligned} w^t &= \sum_{j \in N^t}^{N^t} \alpha_j^t w_j^{t_j} + \left(1 - \sum_{j \in N^t}^{N^t} \alpha_j^t\right) w^{t-1} \\ &= \frac{1}{n^t} \sum_{j \in N_1^{t-1}}^{N_1^{t-1}} n_j w_j^t + \frac{1}{a+1} \frac{1}{n^t} \sum_{i \in N_2^{t-2}}^{N_2^{t-2}} n_i w_i^t + \frac{a(1-z_t)}{a+1} w^{t-1}, \end{aligned} \tag{27}$$

where $\alpha_j^t = \frac{n_j}{n^t} \cdot \frac{1}{a \cdot (t - t_j) + 1}$. t_j refers to local node round of node j and $t - t_j \le 1$. n^t refers to the total sample number of N^t. z_t is the sample proportion of N_1^{t-1} on N^t and $z_t = \frac{1}{n^t} \sum_{j \in N_1^{t-1}}^{N_1^{t-1}} n_j$. So the difference between global loss and optimal loss can be expressed as follows

$$\begin{aligned} G(t) &= \mathcal{L}(w^t) - \mathcal{L}(w^*) \\ &\le \frac{1}{n^t} \sum_{j \in N_1^{t-1}}^{N_1^{t-1}} n_j \mathcal{L}(w_j^t) + \frac{1}{a+1} \frac{1}{n^t} \sum_{i \in N_2^{t-2}}^{N_2^{t-2}} n_i \mathcal{L}(w_i^t) \\ &\quad + \frac{a(1-z_t)}{a+1} \mathcal{L}(w^{t-1}) - \mathcal{L}(w^*) \\ &\le z_t \lambda G(t-1) + \frac{z_t L^2 \eta^2 C_1}{2(L-\mu)} + \frac{(1-z_t)\lambda}{a+1} G(t-2) \\ &\quad + \frac{1-z_t}{a+1} \frac{L^2\eta^2 C_1}{2(L-\mu)} + \frac{a(1-z_t)}{a+1} G(t-1) \\ &\le \left(z_t \left(\lambda - \frac{a}{a+1} \right) + \frac{a}{a+1} \right) G(t-1) \\ &\quad + \frac{(1-z_t)\lambda}{a+1} G(t-2) \\ &\quad + \frac{a z_t + 1}{a+1} \frac{L^2\eta^2 C_1}{2(L-\mu)}. \end{aligned}$$

$$\leq \frac{a}{a+1}G(t-1)+\frac{\lambda}{a+1}G(t-2)+\frac{L^2\eta^2C_1}{2(L-\mu)}$$
$$\leq d_1G(t-1)+s^{t-1}\left[G(1)-d_1G(0)+d_2\right]-d_2, \quad (28)$$

where $\lambda=(1-\frac{\mu}{L})^{\mathcal{B}}$. $d_1=\frac{-2\lambda}{\sqrt{a^2+4\lambda(a+1)}+a}$, $s=\frac{2\lambda}{\sqrt{a^2+4\lambda(a+1)}-a}$ and $d_2=\frac{L^2\eta^2C_1\left(\sqrt{a^2+4\lambda(a+1)}+a+2\right)}{4(\lambda-1)(L-\mu)}$.

APPENDIX B
DERIVATION OF DYNAMIC WAITING TIME

At global round t, node j can participate in aggregation only if $\tau_j^t \geq \tau_s^t$, where τ_s^t refers to server waiting time. According to the definition of (5), we have $\tau_j^t=\tau_{j,c}^t+\tau_{j,com}^t$. We ignore the influence of $\tau_{j,c}^t$ in τ_j^t by setting $\tau_{j,c}^t=C_4$ where C_4 is a constant. So the probability that single vehicle can participate in aggregation follows binomial distribution is expressed as:

$$p(X)=\frac{\pi d_{\max}^2}{S_a/n_a}=\frac{n_a\pi P}{S_aN\eta_0\left(2^{\frac{r_{\min}}{B}}-1\right)}, \quad (29)$$

where $d_{\max}$ and $r_{\min}$ refer to average maximum relative distance and average minimum transmission rate that nodes can participate in aggregation, respectively. For M vehicles, the expected proportion of participating nodes is

$$ar^e=\frac{1}{M}Mp(X)=\frac{n_a\pi P}{S_aN\eta_0\left(2^{\frac{r_{\min}^e}{B}}-1\right)}, \quad (30)$$

where $r_{\min}^e$ is expected average minimum transmission rate, and there are $n_a\cdot ar^e$ vehicles whose average transmission rate at this round is bigger than $r_{\min}^e$. After setting a fixed value of ar^e, the expected server waiting time is

$$\tau_s^e=\frac{\phi}{r_{\min}^e}+C_4=\frac{\phi}{B\log_2\left(1+\frac{n_a\pi P}{S_aN\eta_0\cdot ar^e}\right)}+C_4. \quad (31)$$

However, due to the influence of some uncontrollable factors such as road restrictions and local computation time, the average transmission rate of each global round is not exactly as expected. Therefore, the server waiting time of each round needs to be adjusted dynamically.

Due to the continuity of vehicle movement, we use the aggregating proportion of t round ar^t to predict the expected waiting time change $\Delta\tau^t(ar^t)$. Let $f(x)=\log_2(1+\frac{n_a\pi P}{S_aN\eta_0\cdot x})$, so we have

$$\tau_s^{t+1}=\tau_s^t+\Delta\tau^t(ar^t)=\tau_s^t+\frac{\phi}{B}\left(\frac{1}{f(ar^t)}-\frac{1}{f(ar^e)}\right). \quad (32)$$

Except for ar^t, all variables in (32) are fixed values in simulation environment. For $0<ar^t\leq 1$, the diagram of $\Delta\tau^t(ar^t)$ is shown in Fig. 17, where $ar^e=0.8$.

In Fig. 17, it can be seen that the relationship between $\Delta\tau^t(ar^t)$ and ar^t is linearity within [0,1] by numerical simulation. However, the linear relationship can't be constrained by upper and lower bounds. Because the server waiting time is an important factor that affects the efficiency of federated learning and system stability, drastically changing server waiting time is bad for resource conservation and model convergence. Therefore, we exploit that ar^e-ar^t follows logistic distribution to constrain the maximum and change speed of server waiting time. So we have

$$\tau_s^{t+1}=\tau_s^t+\beta_2\cdot\frac{e^{\beta_1(ar^e-ar^t)}-1}{e^{\beta_1(ar^e-ar^t)}+1}, \quad (33)$$

where β_1 and β_2 are hyperparameter.

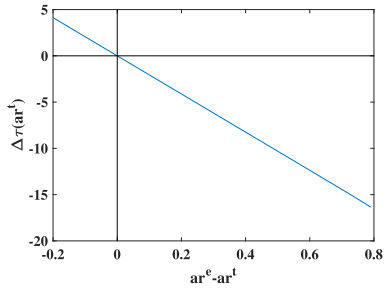


Fig. 17. Relationship between $\Delta\tau^t(ar^t)$ and ar^e-ar^t.

REFERENCES

[1] 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.

[2] A. Fang, X. Peng, J. Zhou, and L. Tang, "Research on the map-matching and spatial-temporal visualization of expressway traffic accident information," in *Proc. 3rd IEEE Int. Conf. Intell. Transp. Eng.*, 2018, pp. 23–27.

[3] Y. Yanbin, Z. Lijuan, L. Mengjun, and S. Ling, "Early warning of traffic accident in Shanghai based on large data set mining," in *Proc. Int. Conf. Intell. Transp., Big Data Smart City*, 2016, pp. 18–21.

[4] C. Zheng, X. Fan, C. Wen, L. Chen, C. Wang, and J. Li, "DeepSTD: Mining spatio-temporal disturbances of multiple context factors for citywide traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3744–3755, Sep. 2020.

[5] Y. Guo, S. Wang, L. Zheng, and M. Lu, "Trajectory data driven transit-transportation planning," in *Proc. 5th Int. Conf. Adv. Cloud Big Data*, 2017, pp. 380–384.

[6] P. N. M. Choudhary, N. R. Pawar Velaga, and D. S. Pawar, "Overall performance impairment and crash risk due to distracted driving: A comprehensive analysis using structural equation modelling," *Transp. Res. Part F, Traffic Psychol. Behav.*, vol. 74, pp. 120–138, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1369847820305064

[7] E. K. Joa Yi and Y. Hyun, "Estimation of the tire slip angle under various road conditions without tire–road information for vehicle stability control," *Control Eng. Pract.*, vol. 86, pp. 129–143, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0967066118303836

[8] D. Kombate *et al.*, "The internet of vehicles based on 5G communications," in *Proc. IEEE Int. Conf. Internet Things IEEE Green Comput. Commun. IEEE Cyber, Phys. Soc. Comput. IEEE Smart Data*, 2016, pp. 445–448.

[9] L. J. Zhu Di and L. Su, "Intelligent connected vehicles 2030: Research and trend analysis of sensor Big Data," in *Proc. Electron. Product Rel. Environ. Testing*, 2020, pp. 93–97.

[10] B. McMahan and D. Ramage, "Federated learning over wireless fading channels," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3546–3557, 2020.

[11] X. Huang, P. Li, R. Yu, Y. Wu, K. Xie, and S. Xie, "FedParking: A federated learning based parking space estimation with parked vehicle assisted edge computing," *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 9355–9368, Sep. 2021.

[12] Z. Zhang, S. Wang, Y. Hong, L. Zhou, and Q. Hao, "Distributed dynamic map fusion via federated learning for intelligent networked vehicles," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 953–959.

[13] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," 2016, *arXiv:1610.02527*.

[14] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Differentially private asynchronous federated learning for mobile edge computing in urban informatics," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 2134–2143, Mar. 2020.
[15] A. Nilsson, S. Smith, G. Ulm, E. Gustavsson, and M. Jirstrand, "A performance evaluation of federated learning algorithms," in *Proc. 2nd Workshop Distrib. Infrastruct. Deep Learn.*, 2018, pp. 1–8.
[16] W. Wu, L. He, W. Lin, R. Mao, C. Maple, and S. A. Jarvis, "SAFA: A semi-asynchronous protocol for fast federated learning with low overhead," *IEEE Trans. Comput.*, vol. 70, no. 5, pp. 655–668, May 2021.
[17] Z. M. Ma, X. Zhao Cai, and Z. Jia, "Fast-convergent federated learning with class-weighted aggregation," *J. Syst. Architecture*, vol. 117, 2021, Art. no. 102125. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S138376212100093X
[18] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *Proc. IEEE Int. Conf. Commun.*, 2019, pp. 1–7.
[19] S. Abdulrahman, H. Tout, A. Mourad, and C. Talhi, "FedMCCS: Multicriteria client selection model for optimal IoT federated learning," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4723–4735, Mar. 2021.
[20] Y. Wang and B. Kantarci, "A novel reputation-aware client selection scheme for federated learning within mobile environments," in *Proc. IEEE 25th Int. Workshop Comput. Aided Model. Des. Commun. Links Netw.*, 2020, pp. 1–6.
[21] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10700–10714, Dec. 2019.
[22] J. Kang, Z. Xiong, D. Niyato, Y. Zou, Y. Zhang, and M. Guizani, "Reliable federated learning for mobile networks," *IEEE Wireless Commun.*, vol. 27, no. 2, pp. 72–80, Apr. 2020.
[23] C. Xie, S. Koyejo, and I. Gupta, "Asynchronous federated optimization," 2019, *arXiv:1903.03934*.
[24] M. B. Chen Mao and T. Ma, "FedSA: A staleness-aware asynchronous federated learning algorithm with non-IID data," *Future Gener. Comput. Syst.*, vol. 120, pp. 1–12, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X21000649
[25] A. Imteaj and M. H. Amini, "Fedar: Activity and resource-aware federated learning model for distributed mobile robots," in *Proc. 19th IEEE Int. Conf. Mach. Learn. Appl.*, 2020, pp. 1153–1160.
[26] Y. Chen, X. Sun, and Y. Jin, "Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 10, pp. 4229–4238, Oct. 2020.
[27] Z. W. Chen, K. Liao, C. Hua Lu, and W. Yu, "Towards asynchronous federated learning for heterogeneous edge-powered Internet of Things," *Digit. Commun. Netw.*, vol. 7, pp. 317–326, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352864821000195
[28] Y. Wang, "CO-OP: Cooperative machine learning from mobile devices," 2017, [Online]. Available: https://era.library.ualberta.ca/items/7d680f04-7987-45c5-b9cd-4fe43c87606f
[29] K. Miller, *Communication Theories*. New York, NY, USA: Macgraw-Hill, 2005.
[30] M. R. Nasr Shokri and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *Proc. IEEE Symp. Secur. Privacy*, 2019, pp. 739–753. [Online]. Available: http://dx.doi.org/10.1109/SP.2019.00065
[31] "OpenStreetMap," Univ. College London, 2019, Accessed: Apr. 15, 2020. [Online]. Available: https://www.openstreetmap.org
[32] Baidu Map, OpenGPS, "OpenGPS location service," 2018, Accessed: 2018. [Online]. Available: https://www.opengps.cn/Data/Cell/Region.aspx
[33] P. A. Lopez *et al.*, "Microscopic traffic simulation using SUMO," in *Proc. 21st IEEE Int. Conf. Intell. Transp. Syst.*, 2018, pp. 2575–2582. [Online]. Available: https://elib.dlr.de/124092/
[34] J. Hyun, Y. Won, E. Kim, J. Yoo, and J. W. Hong, "Is LTE-advanced really advanced?," in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp.*, 2016, pp. 703–707.
[35] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, [Online]. Available: https://arxiv.org/abs/1708.07747
[36] X. Huang, P. Wang, X. Cheng, D. Zhou, Q. Geng, and R. Yang, "The ApolloScape open dataset for autonomous driving and its application," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2702–2719, Oct. 2020.

Feiyuan Liang received the bachelor's degree in traffic engineering in 2019 from Sun Yat-sen University, Guangzhou, China, where he is currently working toward the master's degree with the School of Intelligent Systems Engineering. His research interests include internet of vehicles, federated learning, and autonomous driving.

Qinglin Yang (Member, IEEE) received the B.S. degree from the School of Geographical Sciences, in 2014, the M.S. degree in computing mechanism from the College of Civil Engineering, Kunming University of Science and Technology, Kunming, China, in 2017, and the Ph.D. degree in computer science and engineering from The University of Aizu, Aizuwakamatsu, Japan, in 2021. He is currently a Postdoctoral with the School of Intelligent System Engineering, Sun Yat-sen University, Guangzhou, China. His research interests include edge computing, federated learning, and green computing.

Ruiqi Liu is currently working toward the undergraduate degree with the School of Intelligent System Engineering, Sun Yat-sen University, Guangzhou, China. His research interests include federating learning and Internet of Vehicles.

Junbo Wang (Member, IEEE) received the Ph.D. degree in computer science and engineering from The University of Aizu, Aizuwakamatsu, Japan, in 2011. He was a Postdoctoral and an Associate Professor with The Univeristy of Aizu. He is currently an Associate Professor with the School of Intelligent Systems Engineering, Sun Yat-sen University, Guangzhou, China. He was the PI with Japan site for JST-NSF Joint Funding to study Big Data and disaster (SICORP Project) cooperating with Dr. Krishna Kang and Dr. Amitangshu Pal. His research interests include collaborative ML, federated learning, fog computing, Big Data, and privacy.

Kento Sato (Member, IEEE) is currently a Team Leader of High Performance Big Data Research Team with the Center for Computational Science, RIKEN (RIKEN R-CCS), Tokyo, Japan. His research interests include distributed systems and parallel computing, particularly in high performance computing (HPC). His main research interests include artificial intelligence, machine learning and deep learning in HPC, application reproducibility, scalable fault tolerance, and I/O optimization, co-designing, and cloud computing.

Jian Guo (Member, IEEE) received the Ph.D. degree from the Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Tokyo, Japan, in 2019. He is currently a Lecturer with the Department of Big Data and Data Science, Xi'an University of Finance and Economics, Xi'an, China. His research interests include high performance computing (HPC) and machine learning. His main research focuses on HPC and machine learning-based applications in computational social sciences.