



CSA_FedVeh: Cluster-Based Semi-asynchronous Federated Learning Framework for Internet of Vehicles

Dun Cao¹, Jiasi Xiong¹, Nanfang Lei¹, Robert Simon Sherratt²,
and Jin Wang¹(✉)

¹ Changsha University of Science and Technology, Changsha 410114, China
jinwang@csust.edu.cn

² School of Biomedical Engineering, The University of Reading, Reading RG6 6AY,
UK

Abstract. In Internet of Vehicles (IoV) system, Federated Learning (FL) is a novel distributed approach to processing real-time vehicle data that enables training of shared learning models while ensuring data privacy. However, existing FL still face numerous challenges in IoV. Firstly, the fast convergence with FL models is difficult to achieve due to the high mobility of vehicles and the non-independent identical distribution (Non-IID) among data collected by vehicles. Moreover, the parameter aggregation process of FL incurs significant communication overhead, and the varying computing power of vehicles results in the straggler. To address these issues, this paper proposes a Cluster-based Semi-Asynchronous Federated Learning framework for IoV (CSA_FedVeh). Specifically, we propose a Space-Time and Weight DBSCAN density clustering algorithm (STW-DBSCAN) that relies on both the space-time location and model weight similarities of vehicles. Clustering of vehicles can alleviate the impact of Non-IID data, and the joint training of data vehicles can reduce resource consumption and mitigate the straggler effect. In addition, we adopt a semi-asynchronous FL aggregation mechanism to reduce communication time and improve FL efficiency. Experimental results show that compared with baselines under Non-IID datasets, CSA_FedVeh can reduce the running time by about 24.6% to 60.2%, and reduce communication consumption by 3.4% to 62.07% on MNIST dataset and 1.01% to 68.6% on GTSRD dataset.

Keywords: Internet of vehicles · Federated learning · Cluster · Semi-asynchronous

1 Introduction

With the rapid development of wireless communication and Artificial Intelligence (AI) technologies, Internet of Vehicles (IoV) has emerged as a significant application scenario for 5G and beyond. It is playing a crucial role in the fields

of autonomous driving and Intelligent Transportation Systems (ITS) [1, 2]. However, Intel estimates that each smart car will generate approximately 4000GB of data per day, which is equivalent to the data produced by nearly 3000 mobile phone users. The real-time processing of the data collected from the vehicles poses a thorny issue. Meanwhile, with the development of Mobile Edge Computing (MEC) and Federated Learning (FL) technologies, on the one hand, as the vehicle data is generated in IoV, MEC naturally combines with IoV, enabling data processing to be performed in the vicinity of the vehicles through the computing power and storage resources of edge server (ES) [3]. On the other hand, in 2016, Google proposed federated learning [4] as a distributed deep learning paradigm, which allows vehicles to train their local deep learning models independently using local data and aggregates them into a global model. Vehicles do not directly send local data and only share local model parameters, which to some extent, protects vehicle privacy [5]. As real-time computing services on the vehicular edge continue to grow, the combination of IoV and FL technology will become a research focus.

Although the existing FL clustering approaches and aggregation mechanisms have been effective in some IoV scenarios, several challenges persist in IoV, including: as a result of differences in sensors and processors of vehicles and devices, the data collected by vehicles is the non-independent identical distribution (Non-IID). When using such Non-IID data, FL model may significantly decrease in terms of convergence speed and accuracy [6, 7]. Vehicles are typically in a state of high-speed mobility and their distance to ES varies over time, which can result in communication congestion and delays when participating vehicles of FL frequently update model parameters to ES. Additionally, the computational capabilities of some vehicles differ, resulting in slow-performing stragglers significantly prolonging the delay of each round FL aggregation and ultimately impacting the convergence speed of the global model.

To tackle the aforementioned three challenges, we acknowledge the significance of cooperation among vehicles and propose a novel vehicle clustering-based semi-asynchronous federated learning framework for IoV (CSA_FedVeh). Our contributions are summarized as follows:

- We establish a distributed training network for FL, which combines local training in vehicles and global aggregation in ES. To ensure the quality of FL model for vehicles and support the faster possible model convergence, we formulate a minimization problem for the convergence time of global model aggregation.
- Based on the CSA_FedVeh framework, we propose a Space-Time and Weight DBSCAN density clustering algorithm (STW-DBSCAN) that relies on both the space-time location similarities and model weight similarities of vehicles. This algorithm efficiently solves the straggler problem and accelerates local model training. Meanwhile, a semi-asynchronous federated aggregation mechanism is adopted to further reduce resource consumption and communication costs by adjusting server waiting time.

- We establish a simulation for vehicle-clustered FL network. Experimental results demonstrate that, under a fixed system operation time, our CSA.FedVeh framework outperforms four other benchmark frameworks by shortening the running time by approximately 24.6% to 60.2%, while achieving similar accuracy. Additionally, on MNIST dataset, the communication consumption is reduced by 3.4% to 62.07%, and on GTSRD dataset, the communication consumption is reduced by 1.01% to 68.6%, when compared to achieving similar accuracy.

2 Related Work

In recent years, an increasing number of scholarly investigations have endeavored to implement FL frameworks within IoV scenario [8–10]. Huang *et al.* [8] propose a novel FL framework called ‘FedParking’ that assists parked vehicles in providing computational offloading services and utilizes LSTM model for parking space estimation. Liang *et al.* [9] propose a semi-synchronous FL (Semi-SynFed) protocol and a dynamic aggregation scheme to asynchronously aggregate model parameters, in order to enhance the performance of FL in IoV scenario. Huang *et al.* [10] propose an asynchronous FL privacy-preserving computation model (AFLPC) for 5G-V2X, which aims to better utilize the low latency advantage of 5G networks, while also protecting data privacy in IoV. Similar to the aforementioned framework, we also considered the implementation of FL in IoV for real-time processing of data collected by vehicles.

Existing FL frameworks have been effective in addressing the impact of Non-IID data and resource constraints [11–13]. Ma *et al.* [11] propose a task offloading method based on data and resource heterogeneity in the HFEL environment, incorporating the statistical features of data through information entropy into the cost function to reshape the edge data. Briggs *et al.* [12] improve FL by introducing a hierarchical clustering step (FL+HC), which separates client clustering based on the similarity between clients’ local updates and the global joint model. Tan *et al.* [13] propose a novel federated prototype learning (FedProto) framework, in which communication between devices and servers is done via class prototypes rather than gradients. Considering the presence of Non-IID data and resource constraints in IoV, we propose a clustering algorithm in this paper that alleviates the problem of Non-IID data and resource constraints in IoV, while effectively mitigating the impact caused by high vehicle mobility in IoV.

Since global aggregation is required for parameter uploading in FL, existing FL aggregation mechanisms can be classified into two types based on their aggregation mechanisms: synchronous [4] and asynchronous [14] mechanisms. For synchronous FL mechanisms, the ES needs to collect all the parameters obtained from the participating vehicles before executing the aggregation process. However, the impact of stragglers [15], caused by poor network or hardware resources of some vehicles, can lead to significant delays. As for asynchronous FL mechanisms, the ES can aggregate the parameters without waiting for all vehicles in a round FL aggregation, but this can result in gradient divergence,

further decreasing the performance of the FL model. In this work, we adopt the semi-asynchronous mechanism [16–18], which further reduces resource consumption and communication costs by adjusting the server’s waiting time. Sun *et al.* [17] propose a semi-asynchronous FL framework for extremely heterogeneous devices. Ma *et al.* [18] propose a semi-asynchronous federated learning mechanism called ‘FedSA’ and theoretically prove the convergence of FedSA. In contrast to the aforementioned framework, we consider vehicles of high-speed mobility and combine a semi-asynchronous mechanism with clusters of vehicles.

3 System Model and Problem Formulation

In this section, we firstly introduce the clustered federated learning process in IoV scenario. Sequentially, we describe the cluster-based semi-asynchronous federated learning framework (CSA_FedVeh). Finally, we propose the problem of minimizing the global model training time and formalize it for a better address of the challenges in IoV federated learning.

3.1 Vehicle-Clustered Federated Learning Network

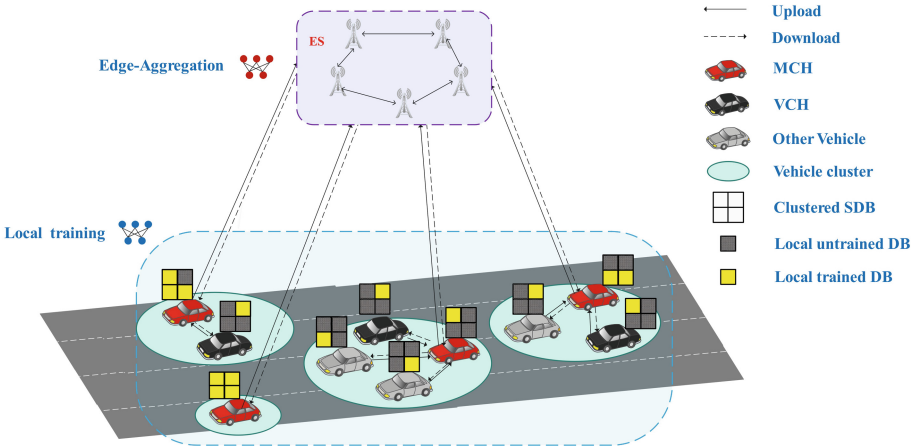


Fig. 1. Illustration of Cluster FL process in IoV.

As shown in the Fig. 1, we consider a vehicle-clustered FL network system, consisting of vehicular users (VUs) and edge server (ES). Assuming N VUs randomly distributed in IoV system, forming a set of VUs $\mathcal{V} = \{1, \dots, n, \dots, N\}$, these VUs are clustered into M vehicle clusters using the STW-DBSCAN algorithm that relies on both the space-time location similarities and model weight similarities of vehicles (introduced in Sect. 4), forming a set of vehicle clusters

$\mathcal{C} = \{c_1, \dots, c_m, \dots, c_M\}$. Assuming convergence of FL model after K rounds of global aggregation, where $k \in \{1, 2, \dots, K\}$.

In the vehicle-clustered FL network system, due to the close proximity of the VUs within a vehicle cluster, the collected information and the trained models also are highly similar, it can be assumed that the data of the VUs in the vehicle cluster are the same and can be partitioned into shared data blocks (SDBs). During the training process, VUs in the vehicle cluster only need to train their own models using their own historical experience data blocks (DBs), without the need to transmit local data, where DBs are partitioned according to the computing capabilities of VUs within each vehicle cluster. The main vehicle cluster head (MCH) is responsible for uploading and downloading and sending model weight parameters. If the MCH is offline, a vice vehicle cluster head (VCH) will be activated to take its place (Table 1).

Table 1. Notations and their meanings.

Notation	Meaning
\mathcal{V}	The set of vehicular users
\mathcal{C}	The set of vehicle clusters
K	The global model will converge after K rounds of global aggregation
C_k	The vehicular users participating in the global updating in round k
q	The number of vehicle clusters participating in each round of global aggregation
$T_{v,n}^{\text{soj}}$	The sojourn time of vehicular user n at the current edge server
$T_{v,n}^{\text{loc}}$	The computation time of each round of local model training for vehicular user n
$T_{c,m}^{\text{loc}}$	The computation time of each round of local model training for vehicle cluster c_m
$T_{k,m}^{\text{comm}}$	The communication time for vehicle cluster c_m in the k -th round of global aggregation
$T_{k,m}^{\text{comp}}(s_k^m)$	The local computation time of vehicle cluster c_m between the start time of the k -th round of global aggregation and the end time of local training
T_k^{sort}	Training time for all clusters sorted in the k -th round of global aggregation
$w_k^{v,n}$	The local model weight parameter derived by local updating on vehicular user n at the k -th round of global aggregation
$w_k^{c,m}$	The cluster model weight parameter derived by cluster updating on vehicle cluster c_m at the k -th round of global aggregation
w_k	The global model weight parameter in round k
$\nabla F_{v,n}(w_k)$	Gradient of vehicular user n at the k -th round of global aggregation
$\nabla F_{c,m}(w_k)$	Gradient of vehicle cluster c_m at the k -th round of global aggregation
$\nabla F(w_k)$	Global gradient of the k -th round of global aggregation
s_k^m	The number of global aggregation rounds that differ between the k -th round of global aggregation in which vehicle cluster c_m participated and the global aggregation in which this c_m participated last time
δ	The model weight parameter of similarity threshold
μ	The global aggregation stopping threshold
τ	The number of local model update rounds for all vehicle users when participating in a global aggregation
ε	The neighborhood threshold for clustering
$N_\varepsilon(n)$	The ε -neighborhood of vehicular user n
$N_\varepsilon^+(n)$	The ε +-neighborhood of vehicular user n

We consider an unidirectional, straight, multi-lane IoV scenario, where VUs travel along the X -axis in the direction of the arrow. At time t , assuming that VU n is traveling at a constant speed \bar{v}_n , its position can be denoted as $\{x_n(t), y_n(t)\}$. The associated ES e is located at a fixed position $\{x_e, y_e\}$ with a coverage radius of r . Therefore, the remaining distance of VU n within the coverage area of its ES can be defined as:

$$\Pi_i = \sqrt{r^2 - (y_e - y_n(t))^2} - (x_n(t) - x_e). \quad (1)$$

Only when VU n is within the coverage area of ES, the parameters can be uploaded to the current ES. Therefore, the sojourn time of VU n at the current ES is defined as:

$$T_{v,n}^{\text{soj}} = \frac{\Pi_n}{\bar{v}_n}, \quad (2)$$

where \bar{v}_n is the speed of VU n .

In order to calculate the distance between arbitrary nodes i and j (including VUs and ES), the Euclidean distance formula is introduced:

$$\text{dist}(i, j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \quad (3)$$

We resort to the Shannon capacity formula to compute the data rate of nodes i to j in FL the k -th round of global aggregation and denoted as:

$$R_{i,j}^k(\text{dist}(i, j)) = B_i \log_2 \left(1 + \frac{P_i^{\text{tx}} \cdot h(\text{dist}(i, j))}{N_0} \right), \quad (4)$$

where N_0 represents the noise power, $h(\text{dist}(i, j))$ is the channel gain at the distance between node i and j , P_i^{tx} and B_i represent the transmission power and the communication bandwidth from node i to node j .

During the k -th round of global aggregation, the uplink transmission time for a vehicle cluster c_m to transmit its cluster model weight parameters to its corresponding ES can be expressed as:

$$T_{k,m}^{\text{tx}} = \frac{|w_m^k|}{R_{m,e}^k(\text{dist}(m, e))}. \quad (5)$$

The downlink transmission time for an ES to transmit global model weight parameters to the MCH of vehicle cluster c_m in its coverage area can be expressed as:

$$T_{k,m}^{\text{rx}} = \frac{|w_k|}{R_{e,m}^k(\text{dist}(e, m))}. \quad (6)$$

As the time for intra-cluster transmission of parameters from cluster member VUs to the MCH is short, this transmission time is neglected in this paper. Therefore, for the k -th round of global aggregation in ES, the communication time for a vehicle cluster c_m consists of the uplink transmission time and downlink transmission time of its MCH.

$$T_{k,m}^{\text{comm}} = T_{k,m}^{\text{tx}} + T_{k,m}^{\text{rx}}. \quad (7)$$

3.2 Cluster-Based Semi-asynchronous Federated Learning Framework for IoV (CSA_FedVeh)

In the CSA_FedVeh, assuming \mathcal{C}_k represents the set of vehicle clusters participating in the k -th round of global aggregation, while semi-asynchronous aggregation quantity q is the number of vehicle clusters taking part in each round of global aggregation.

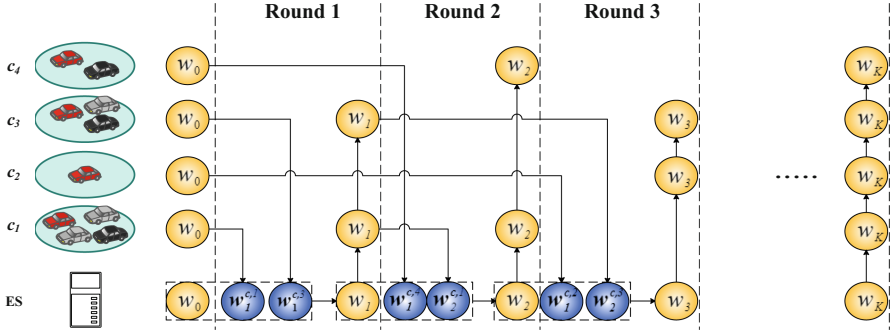


Fig. 2. Illustration of the CSA_FedVeh framework when $q=2$.

The vehicle cluster c_m that participate in global aggregation process download the global training model weight parameters w_k from the ES and distribute them to all VUs within the vehicle cluster. Each VU then updates its local model weight parameters based on its own historical data after uploads their new model weight parameters to the MCH, which synchronously aggregates vehicle cluster model weight parameters $w_k^{c,m}$ and sends them back to the ES. The ES collects q sequentially arrived cluster model weight parameters and performs global aggregation. Meanwhile, the vehicle clusters that did not participate in global aggregation continue their training. After the ES performs global aggregation, it generates the next round of global model weight parameters w_{k+1} and sends them to the MCHs that uploaded local weight parameters in the previous round, which then transmit them to the VUs within their respective vehicle clusters for the next round of training.

For instance, as shown in Fig. 2, The vehicle clusters participating in global aggregation in the first, second, and third rounds are $\mathcal{C}_1 = \{c_1, c_3\}$, $\mathcal{C}_2 = \{c_4, c_1\}$, and $\mathcal{C}_3 = \{c_2, c_3\}$, respectively.

To formalize the problem, we will introduce the CSA_FedVeh framework from three aspects: vehicle user model training, intra-cluster model aggregation, and semi-asynchronous global model aggregation, which is formally described in Algorithm 1.

Vehicle User Model Training. Local loss function: each VU trains a local model based on local DBs, where the loss function of the k -th round FL training model of VU n is defined as:

$$F_{v,n}(w_k^{v,n}) = \frac{1}{D_n} \sum_{(x_d, y_d) \in \mathcal{D}_n} f(w_k^{v,n}, x_d, y_d), \quad (8)$$

where $f(w_k^{v,n}, x_d, y_d)$ is the loss function of the model based on the training set samples x_d and their predicted labels y_d under the local weight parameter $w_k^{v,n}$, \mathcal{D}_n and $D_n \triangleq |\mathcal{D}_n|$ denote the local trained DB and the number of samples of VU n after partitioning the SDB.

Local model update: after receiving the global model weight parameter w_k , VU n performs τ iterations for local parameter updates:

$$w_{k+1}^{v,n} \leftarrow w_k - \eta \nabla F_{v,n}(w_k), \quad (9)$$

where η is the learning rate and $\nabla F_{v,n}(w_k)$ is the gradient computed by local model of VU n under the global weight parameter w_k .

Local resource cost: the computation time and energy consumption of each round of local model training for VU n are denoted as:

$$T_{v,n}^{\text{loc}} = \frac{\sum_{d_n=1}^{D_n} \psi_{d_n}}{c_n^p \cdot f_n}, E_{v,n}^{\text{loc}} = p_n \cdot T_{v,n}^{\text{loc}}, \quad (10)$$

where ψ_{d_n} is the total number of Floating Point Operations per Second (FLOPS) required for sample d_n , VU is characterized by a processing capability equal to c_n^p (FLOPS) per CPU cycle, f_n is CPU frequency and p_n is the computational power of VU.

Intra-Cluster Model Aggregation. Cluster model aggregation: The VUs within vehicle cluster c_m pass the trained local model weight parameter $w_k^{v,n}$ to the MCH for intra-cluster aggregation, obtaining the cluster model weight parameter:

$$w_k^{c,m} \leftarrow \sum_{n \in c_m} D_n \cdot w_k^{v,n}, \quad (11)$$

Cluster gradient aggregation: in order to determine the convergence of the global model, it is necessary to upload the gradients from each vehicle cluster to ES. Therefore, the aggregation of gradients from vehicle cluster c_m is defined as:

$$\nabla F_{c,m}(w_k) \leftarrow \sum_{n \in c_m} \nabla F_{v,n}(w_k). \quad (12)$$

Cluster resource cost: the computation time and energy consumption for training each round of local model in vehicle cluster c_m is defined as:

$$T_{c,m}^{\text{loc}} = \max_{c \in c_m} \{T_{v,c}^{\text{loc}}\}, E_{c,m}^{\text{loc}} = \sum_{c \in c_m} E_{v,c}^{\text{loc}}. \quad (13)$$

Algorithm 1. CSA_FedVeh

Input: the set of vehicle clusters \mathcal{C} , semi - asynchronous aggregation parameters q , number of vehicle local iterations τ

Output: the final global model weight parameter w^*

- 1: Initialize the cluster structure;
- 2: Initialize $w_0, k = 0, T = 0, E = 0$;
- 3: **while** $k \neq K$ and $\|\nabla F(w_k)\| > \mu \|\nabla F(w_{k-1})\|$ and $T < T_{\max}$ and $E < E_{\max}$ **do**
- 4: **Global Aggregation at the Edge Server**
- 5: set $k \leftarrow k + 1$, and $\mathcal{C}_k = \emptyset$;
- 6: **while** $|\mathcal{C}_k| \neq q$ **do**
- 7: Receive update c_m from \mathcal{C} ;
- 8: Compute $T_{k,m}^{\text{total}}$ according to Eq.(18);
- 9: **if** $T_{k,m}^{\text{total}} \leq T_{c,m}^{\text{soj}}$ **then**
- 10: update $\mathcal{C}_k \leftarrow \mathcal{C}_k + \{c_m\}$, $E \leftarrow E + E_{c,m}^{\text{loc}}$;
- 11: **end if**
- 12: **end while**
- 13: Compute $w_k, \nabla F(w_k), T_k$ according to Eq.(14),Eq.(15),Eq.(16);
- 14: Update $T \leftarrow T + T_k, w^* \leftarrow w_k$;
- 15: Send w_k back to \mathcal{C}_k ;
- 16: **Cluster Aggregation at c_m**
- 17: Receive local updates from all VUs in vehicle cluster c_m ;
- 18: Update $T_{c,m}^{\text{soj}} \leftarrow \min_{n \in c_m} \{T_{v,n}^{\text{soj}}\}$;
- 19: Compute $w_k^{c,m}, \nabla F_{c,m}(w_k), T_{c,m}^{\text{loc}}$ and $E_{c,m}^{\text{loc}}$ according to Eq.(11),Eq.(12),Eq.(13);
- 20: Send $w_k^{c,m}, \nabla F_{c,m}(w_k), T_{c,m}^{\text{soj}}$ and resource consumption $T_{c,m}^{\text{loc}}, E_{c,m}^{\text{loc}}$ to ES;
- 21: Receive w_k from ES and return it back to all VUs in vehicle cluster c_m ;
- 22: **Procedure at Vehicle User n in Cluster c_m**
- 23: Receive w_k from c_m ;
- 24: Perform local updates;
- 25: Compute $T_{v,n}^{\text{loc}}$ and $E_{v,n}^{\text{loc}}$, $T_{v,n}^{\text{soj}}$ according to Eq.(10),Eq.(2);
- 26: Send $w_{k+1}^{v,n}, \nabla F_{v,n}(w_k), T_{v,n}^{\text{soj}}$ and resource consumption $T_{v,n}^{\text{loc}}, E_{v,n}^{\text{loc}}$ to MCH;
- 27: **end while**
- 28: **return** w^* ;

Semi-asynchronous Global Model Aggregation. Global model aggregation: during a round of global aggregation, when ES receives cluster model weight parameters transmitted by q vehicle clusters, ES performs global aggregation:

$$w_{k+1} \leftarrow \frac{1}{D_k} \sum_{m \in \mathcal{C}_k} w_k^{c,m}, \quad (14)$$

where D_k is the total number of samples from all participating vehicle clusters in the k -th round of global aggregation.

Global Gradient aggregation: in order to determine the stopping criterion for the convergence of the global model [19], the global gradient is defined as:

$$\nabla F(w_{k+1}) \leftarrow \frac{1}{q} \sum_{m \in \mathcal{C}_k} \nabla F_{c,m}(w_k). \quad (15)$$

For the k -th round of FL aggregation in ES, the time taken for global aggregation is determined by the longest computation time of the participating vehicle cluster \mathcal{C}_k , denoted as:

$$T_k = \mathcal{T}_k^{sort}[q], \quad (16)$$

where $\mathcal{T}_k^{sort}[q]$ represents the q -th element in the set \mathcal{T}_k^{sort} (defined in Eq.(20)).

Assuming that all VUs update their local weight parameters τ times during each cluster weight parameter upload, the local computation time of vehicle cluster c_m is the duration between the start time of the k -th round of global aggregation and the end time of local training, denoted as:

$$T_{k,m}^{comp}(s_k^m) = \begin{cases} T_{c,m}^{loc} \cdot \tau - \sum_{k-s_k^m}^{k-1} T_k, & s_k^m > 0 \\ T_{c,m}^{loc} \cdot \tau, & s_k^m = 0 \end{cases}, \quad (17)$$

where s_k^m represents the number of global aggregation rounds that differ between the k -th round of global aggregation in which vehicle cluster c_m participated and the global aggregation in which this vehicle cluster c_m participated last time, for example in Fig. 2, where $s_1^3 = 0$, $T_{1,3}^{comp}(s_1^3) = T_3^{loc} \cdot \tau$; $s_2^1 = 0$, $T_{2,1}^{comp}(s_2^1) = T_1^{loc} \cdot \tau$; $s_3^2 = 2$, $T_{3,2}^{comp}(s_3^2) = T_2^{loc} \cdot \tau - \sum_{3-2}^{3-1} T_k = T_2^{loc} \cdot \tau - T_2 - T_1$.

In the k -th round of ES global aggregation, the total time consumption of vehicle cluster c_m is the sum of local computation time and communication time, represented as:

$$T_{k,m}^{total} = T_{k,m}^{comp}(s_k^m) + T_{k,m}^{comm}. \quad (18)$$

Assuming that all vehicle clusters participate in the k -th round of global aggregation, and the set of training times for all vehicle clusters \mathcal{C} in the k -th round of global aggregation is defined as:

$$\mathcal{T}_k = \{T_{k,1}^{total}, \dots, T_{k,m}^{total}, \dots, T_{k,M}^{total}\}. \quad (19)$$

We sort the set of training times \mathcal{T}_k in ascending order:

$$\mathcal{T}_k^{sort} = sort(\mathcal{T}_k). \quad (20)$$

We define the matrix of the number of times each vehicle cluster participates in the global aggregation of ES as:

$$G = (g_1 \dots g_m \dots g_M), \quad (21)$$

where g_m denotes the number of times that vehicle cluster c_m participates in the global aggregation of ES, $0 \leq g_m \leq K$.

We define the matrix of energy consumption of all vehicle cluster as follows:

$$E = (E_{c,1}^{loc} \dots E_{c,m}^{loc} \dots E_{c,M}^{loc}). \quad (22)$$

3.3 Problem Formulation

The optimization problem formulated by the vehicle-clustered federated learning network and the CSA_FedVe framework can be described as:

$$\begin{aligned}
 \text{(P1)} : & \min_{q, \mathcal{C}} \sum_{k=1}^K T_k \\
 & \left\{ \begin{array}{l}
 C_1 : \|\nabla F(w_{k+1})\| \leq \mu \|\nabla F(w_k)\| \\
 C_2 : \sum_{k=1}^K T_k \leq T_{\max} \\
 C_3 : \tau \cdot G \cdot E^T \leq E_{\max} \\
 C_4 : T_{k,m}^{\text{total}} \leq \min_{c \in c_m} \{T_{v,c}^{\text{soj}}\} \\
 C_5 : q \in \{1, 2, \dots, M\}.
 \end{array} \right. \quad (23)
 \end{aligned}$$

In problem P1, the objective function is to minimize the FL training time while satisfying the constraints under the semi-asynchronous aggregation quantity q and the vehicle network clustering strategy \mathcal{C} . Constraint C_1 corresponds to the global aggregation stopping condition [19], where μ ($0 \leq \mu \leq 1$). When $\mu = 0$, the global model achieves a precise solution, whereas $\mu = 1$ indicates that no progress has been made by the global model. Constraint C_2 represents the global training time constraint of FL, and constraint C_3 represents the global training energy constraint of FL. Here, T_{\max} and E_{\max} refer to the maximum acceptable global training time and energy consumption of FL, respectively. Constraint C_4 denotes the total time spent on the k -th round of global aggregation of vehicle cluster c_m must not exceed the minimum sojourn time of VUs in the vehicle cluster c_m .

As many machine learning models have complex intrinsic properties, it is difficult to find closed-form solutions for the objective function. Therefore, in Sect. 4, we design some novel solutions that reduce the training time and resource costs of FL while maintaining learning accuracy.

4 Methodology

To solve the aforementioned problems, in this section, we propose the STW-DBSCAN clustering algorithm, which is designed to determine vehicle clustering strategies in dynamic IoV system. The algorithm decomposes the originally high-complex intrinsic properties problem into two sub-problems: the vehicle clustering problem and the semi-asynchronous aggregation mechanism problem, in order to approximate the solution to the original problem while mitigating the impact of Non-IID data on the FL process.

4.1 STW-DBSCAN Density Clustering Algorithm

In order to reduce the complexity of solving P1, we propose a density clustering algorithm (STW-DBSCAN) that relies on both the space-time location similarities and model weight similarities of vehicles to determine the clustering strategy

Algorithm 2. STW-DBSCAN

Input: the set of all VUs \mathcal{V} , neighborhood parameters $(\varepsilon, MinPts)$, similarity threshold $\delta(2\%$, default)

Output: $\mathcal{C} = \{c_1, \dots, c_m, \dots, c_M\}$

- 1: Initialize the vehicle structure;
- 2: Initialize candidate set $\Omega = \emptyset$, number of clusters $m = 0$, sets of unvisited vehicles $\Gamma = \mathcal{V}$, vehicle network clustering strategy $\mathcal{C} = \emptyset$;
- 3: **for** n in \mathcal{V} **do**
- 4: compute $N_\varepsilon(n)$ and $N_\varepsilon^+(n)$ according to Eq.(24) and (26);
- 5: **if** $|N_\varepsilon^+(n) \cap N_\varepsilon(n)| \geq MinPts$ **then**
- 6: $\Omega = \Omega \cup \{n\}$;
- 7: **end if**
- 8: **end for**
- 9: **while** $\Omega \neq \emptyset$ **do**
- 10: $\Gamma_{old} = \Gamma$;
- 11: random selection of a candidate $o \in \Omega$, initializing queue $Q = \langle o \rangle$;
- 12: $\Gamma = \Gamma \setminus \{o\}$;
- 13: **while** $Q \neq \emptyset$ **do**
- 14: fetch the first sample q in the queue Q ;
- 15: **if** $|N_\varepsilon^+(q)| \geq MinPts$ **then**
- 16: $\Delta = N_\varepsilon^+(q) \cap \Gamma$;
- 17: add the samples in Δ to the queue Q ;
- 18: $\Gamma = \Gamma \setminus \Delta$;
- 19: **end if**
- 20: **end while**
- 21: $m = m + 1$, generate clusters $c_m = \Gamma_{old} \setminus \Gamma$;
- 22: select a random benchmark $\alpha \in c_m$;
- 23: **for** β in c_m **do**
- 24: **if** $\theta(w_1^{v,\alpha}, w_1^{v,\beta}) \leq \delta$ **then**
- 25: $c_m \setminus \beta$;
- 26: **end if**
- 27: **end for**
- 28: $\Omega = \Omega \setminus c_m$, $\mathcal{C} = \mathcal{C} \cup c_m$;
- 29: **end while**
- 30: **return** $\mathcal{C} = \{c_1, \dots, c_m, \dots, c_M\}$;

in problem P1. The algorithm integrates the space-time location constraints into the DBSCAN [20] clustering algorithm to guarantee that VUs within the ES region stay within the range of the vehicle cluster. Additionally, considering the Non-IID of data collected by vehicles, the cosine similarity between VU model weight parameters is calculated to ensure that the vehicle data within a cluster belongs to the same distribution [12]. Lastly, The MCH is chosen based on the latest sojourn time, while the VCH is chosen based on the second latest sojourn time. Based on the algorithm, we can acquire the set of vehicle clusters \mathcal{C} , and by combining \mathcal{C} with Eq.(2), we can compute MCH and VCH.

The STW-DBSCAN algorithm is mainly determined by the parameters of neighborhood threshold ε , density threshold $MinPts$, vehicle sojourn time T_n^{soj} ,

vehicle speed \bar{v} , model weight parameter set of VU, and similarity threshold δ , where ε and $MinPts$ are system-determined hyperparameters. For a VU $n \in \mathcal{V}$, we select the set of VUs in the set \mathcal{V} whose distance from VU n does not exceed ε , as the ε -neighborhood of the VU n ($N_\varepsilon(n)$), denoted as:

$$N_\varepsilon(n) = \{p \in \mathcal{V} | dist(n, p) \leq \varepsilon\}. \quad (24)$$

We determine the sojourn time among the VUs in the set $N_\varepsilon(n)$ by using Eq.(2) and compare it to determine the minimum sojourn time in $N_\varepsilon(n)$:

$$T_{N,n}^{soj} = \min_{p \in N_\varepsilon(n) + \{n\}} \{T_{v,p}^{soj}\}. \quad (25)$$

In IoV system, if the distance between the VU n and all other VUs in $N_\varepsilon(n)$ still satisfies within ε after the minimum sojourn time $T_{N,n}^{soj}$, then it is referred to as the ε +-neighborhood of the VU n ($N_\varepsilon^+(n)$) and denoted as:

$$N_\varepsilon^+(n) = \{p \in N_\varepsilon(n) | \sqrt{((x_n - \bar{v}_n \cdot T_{N,n}^{soj}) - (x_p - \bar{v}_p \cdot T_{N,n}^{soj}))^2 + (y_n - y_p)^2} \leq \varepsilon\}. \quad (26)$$

If $N_\varepsilon^+(n)$ has at least $MinPts$ other VUs and denoted as:

$$|N_\varepsilon^+(n)| \geq MinPts, \quad (27)$$

then vehicle cluster c_m is created, and VU n and all VUs in $N_\varepsilon^+(n)$ are added to the cluster, and all VUs in $N_\varepsilon^+(n)$ are added to the candidate set Ω . Each VU $o \in \Omega$ is checked in turn to see if $N_\varepsilon^+(o)$ contains at least $MinPts$ other VUs, if o has not been added to the vehicle cluster yet, it is added to vehicle cluster c_m , and o is removed from the candidate set Ω , and $N_\varepsilon^+(o)$ is added to the candidate set Ω . This process continues until $\Omega = \emptyset$. Additionally, if VUs that have not been clustered what contain at least $MinPts$ other VUs in their ε +-neighborhood, a new vehicle cluster and candidate set are created.

Randomly select a VU $\alpha \in c_m$ as a baseline, and calculate the cosine similarity of the model weight parameters between VU α and all other VUs $\beta \in c_m$,

$$\theta(w_1^{v,\alpha}, w_1^{v,\beta}) = \frac{(w_1^{v,\alpha})^T w_1^{v,\beta}}{\|w_1^{v,\alpha}\| \|w_1^{v,\beta}\|}, \alpha, \beta \in c_m, \quad (28)$$

judge whether the data distribution of VUs in vehicle cluster c_m is similar. If $\theta(w_1^{v,\alpha}, w_1^{v,\beta}) \leq \delta$, indicating low similarity, remove β from vehicle cluster c_m and re-cluster β , where δ ($-1 \leq \delta \leq 1$) is the similarity threshold, and the closer δ is to 1, the more similar the data distribution is, while the closer it is to -1, the less similar it is. The STW-DBSCAN algorithm is formally described in Algorithm 2.

4.2 Semi-asynchronous

Once the clustering strategy \mathcal{C} of STW-DBSCAN algorithm is fixed, P1 is redefined as a problem of solving a single variable q :

$$\begin{aligned} \text{(P2)} : \min_q \sum_{k=1}^K T_k \\ \text{s.t. } C_1, C_2, C_3, C_5. \end{aligned} \quad (29)$$

We adopt a semi-asynchronous aggregation mechanism [16] to accelerate the global model training speed. Each time global aggregation selects q cluster model weight parameters that arrive in order for aggregation.

5 Performance Evaluation

5.1 Simulation Setting

Benchmarks. We utilize three classic FL frameworks and a semi-asynchronous framework with randomized clustering as benchmarks for performance comparison.

- FedAF: FedAF (FedAvgfull) is a synchronous FL framework, which is a variant of FedAvg [4]. In FedAF framework, all VUs participate in the global updating in each round.
- FedASY [14]: FedASY is an asynchronous FL framework, where ES immediately performs a global updating upon receiving local model weight parameters from any VU.
- SAFA [16]: SAFA is a semi-asynchronous FL framework. For simplicity, the client selection in SAFA is removed, and we naturally set it to half of the total number of VUs in our experiment, simulating the framework of semi-asynchronous aggregation under the condition of no clustering.
- R-SAFA: R-SAFA adopts the SAFA aggregation framework with K-means [21] random clustering, where K is set to be the same as the number of clusters in STW-DBSCAN, simulating the framework of semi-asynchronous aggregation under the condition of general clustering.

Models and Datasets. In order to ascertain the efficacy of CSA_FedVeh framework, we conducted experiments using two disparate training models (LR [22] and CNN [23]), and on two real-world datasets (MNIST [24] and German Traffic Sign Recognition Database (GTSRD) [25]). MNIST dataset comprises 60,000 training samples and 10,000 testing samples, each of which is a grayscale image of a handwritten digit measuring 28×28 pixels. GTSRD dataset includes 43 classes of RGB three-channel traffic sign images, divided into 39,209 training images and 12,630 testing images.

Performance Metrics. We have utilized four commonly-used performance metrics to evaluate the training performance, including:

- 1) Loss Function: used to measure the difference between predicted values and actual values.
- 2) Accuracy: indicates the proportion of correctly classified samples by the model among all samples in the dataset.
- 3) Runtime: indicates the time taken to complete the training process, used to measure the training speed of the model.
- 4) Communication Cost: represents the total communication time spent between all vehicles and the ES upon completion of the training process.

Data Distribution. Considering the heterogeneity of data distribution and the similarity of data collected from vehicles within a certain area among VUs in a real-world IoV scenarios, it is necessary to form a Non-IID dataset among VUs. To achieve data re-distribution, a mixed distribution based on Dirichlet distribution was applied [26], which highlights the similarity of data within the cluster and Non-IID among VUs. The parameters for the mixed distribution were set as $a = 1.0$ and $n = 3$.

Simulation Parameters. In the simulation of IoV scenarios for FL, we consider 50 VUs participating, with safe distances randomly scattered along the lane. The average vehicle speed is approximately 43.6km/h, and the vehicles are given random speeds. We set the neighborhood threshold $\varepsilon=25\text{m}$ and the density threshold $MinPts=1$, with similarity threshold δ of 2%. We use the same $batchsize = 64$ for all VUs. The global learning rate is set to $\eta=0.01$ for both MNIST and GTSRD, and the number of local updates per epoch is set to $H=30$.

5.2 Simulation Results

In this section, we compared our CSA_FedVeh framework with the baseline by training models for 30000s and 50000s on Non-IID MNIST and GTSRD datasets, respectively. Finally, Table 2 lists more detailed training performance comparisons on MNIST and GTSRD datasets. The results show that CSA_FedVeh framework can work better even when the data distribution is Non-IID.

Table 2. Performance comparison of CSA_FedVeh with four benchmarks under two models.

Dataset-Model	Performance Metrics	FedAF	FedASY	SAFA	R-SAFA	CSA_FedVeh
MNIST-LR	Accuracy	91.8%	92.5%	92.2%	93.2%	94.5%
	Runtime for Loss=0.5	7381s	6060s	6458s	4857s	3348s
	Communication time for accuracy=90%	8.35s	18.352s	13.325s	7.04s	6.96s
MNIST-CNN	Average communication rounds	305	826.38	549	579	501.76
	Accuracy	97%	97.7%	97.6%	97.8%	98.4%
	Runtime for Loss=0.5	5412s	3230s	3785s	3276s	2348s
GTSRD-LR	Communication time for accuracy=90%	4.75s	7.55s	6.075s	3.9424s	3.808s
	Accuracy	76.4%	81.8%	79.9%	86%	89.3%
	Runtime for Loss=1	48027s	36749s	40852s	25689s	19077s
GTSRD-CNN	Communication time for accuracy=75%	24.05s	49.375s	36.275s	15.76s	15.6s
	Average communication rounds	509	1377.7	916	965.5	838.4
	Accuracy	88%	91.7%	90.3%	95.5%	96.1%
GTSRD-CNN	Runtime for Loss=1	28934s	23014s	25318s	15751s	11868s
	Communication time for accuracy=75%	16.2s	33.476s	24.375s	11.352s	10.496s

Convergence Performance. In MNIST dataset, as shown in Fig. 3(e), when the global model training loss of LR model drops to 0.5, CSA_FedVeh reaches its fastest runtime of 3348 s, which is 44.7% faster than FedASY, 54.6% faster than FedAF, 48.1% faster than SAFA, and 31% faster than R-SAFA. In Fig. 3(f), when the global model training loss of CNN model drops to 0.5, CSA_FedVeh reaches its fastest runtime of 2348 s, which is 27.3% faster than FedASY, 56.6% faster than FedAF, 37.9% faster than SAFA, and 28.3% faster than R-SAFA. In GTSRD dataset, as shown in Fig. 3(g), when the global model training loss value reached 1.0, CSA_FedVeh reached the fastest running time at 19077 s, which was 48% faster than FedASY, 60.2% faster than FedAF, 53.3% faster than SAFA, and 25.7% faster than R-SAFA. As shown in Fig. 3(h), when CNN global model training loss value reached 1.0, CSA_FedVeh reached the fastest running time at 11868 s, which was 48.4% faster than FedASY, 58.9% faster than FedAF, 53.1% faster than SAFA, and 24.6% faster than R-SAFA. Results on different datasets and models indicate that CSA_FedVeh reduces the time required to reach the same loss value by about 24.6%-60.2%.

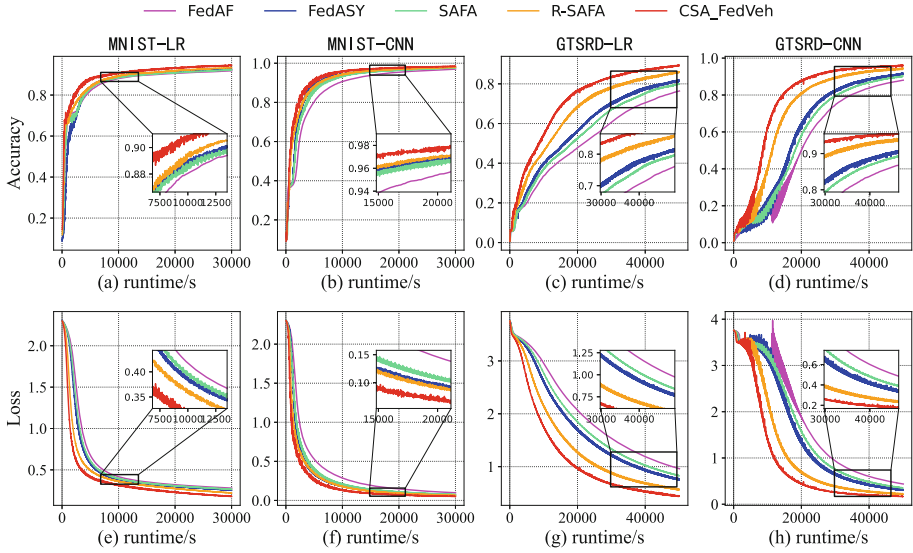


Fig. 3. Accuracy and Loss vs. Runtime with LR and CNN over MNIST and GTSRD.

Not only can CSA_FedVeh stabilize the convergence of the global model, but it also outperforms the four benchmarks in terms of accuracy and convergence speed. Additionally, when comparing Figs. 3(e) and 3(f), as well as Figs. 3(g) and 3(h), CSA_FedVeh exhibits faster convergence speed on CNN models compared to LR models.

Resource Constraints. In MNIST dataset, as shown in Fig. 3(a), when the running time constraint under LR model is set at 30,000s, CSA_FedVeh achieved the highest accuracy of 94.5%. This is 2% higher compared to FedASY, 2.7% higher compared to FedAF, 2.2% higher compared to SAFA, and 1.3% higher compared to R-SAFA. As shown in Fig. 3(b), under CNN model with a running time constraint of 30,000s, CSA_FedVeh achieved the highest accuracy of 98.4%. This is 0.7% higher compared to FedASY, 1.4% higher compared to FedAF, 0.8% higher compared to SAFA, and 0.6% higher compared to R-SAFA. In GTSRD dataset, as shown in Fig. 3(c), under a time constraint of 50000s for LR model, CSA_FedVeh achieved the highest accuracy of 89.3%, which is 7.5% higher than FedASY, 12.9% higher than FedAF, 9.4% higher than SAFA, and 3.3% higher than R-SAFA. As shown in Fig. 3(d), under a time constraint of 50000s for CNN model, CSA_FedVeh achieved the highest accuracy of 96.1%, which is 4.4% higher than FedASY, 8.1% higher than FedAF, 5.8% higher than SAFA, and 0.6% higher than R-SAFA. The results indicate that, under the same time budget, CSA_FedVeh achieved higher accuracy and lower loss than FedAF, FedASY, SAFA, and R-SAFA. This means that CSA_FedVeh can achieve good performance in terms of the balance between convergence speed and accuracy.

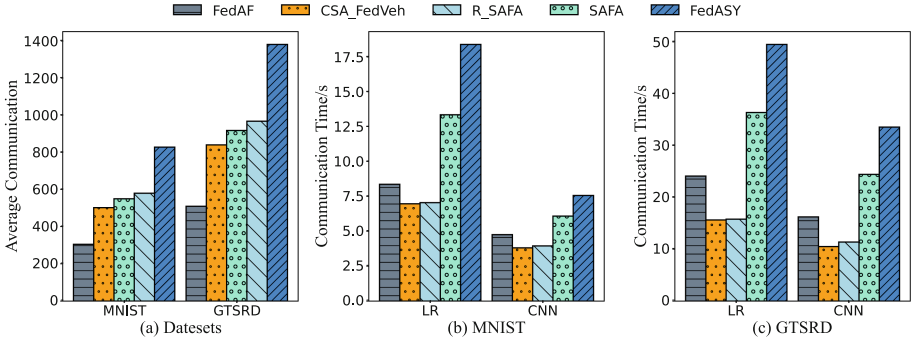


Fig. 4. The Comparison of Communication Resource Consumption. (a) Average Communication rounds. (b) Communication time cost at 90% accuracy of MNIST. (c) Communication time cost at 75% accuracy of GTSRD.

According to Fig. 4(a), on both MNIST and GTSRD datasets, CSA_FedVeh has an average number of communication rounds of 501.76 and 838.4, respectively. This is lower than FedASY by 324.62 and 539.3 rounds, lower than SAFA by 47.24 and 77.6 rounds, lower than R-SAFSA by 77.24 and 127.1 rounds, and higher than FedAF by 196.76 and 329.4 rounds. These results indicate that CSA_FedVeh has a lower average number of communication rounds than FedASY, SAFA, and R-SAFSA, and is second only to FedAF in this respect. Moreover, CSA_FedVeh achieves a higher accuracy than FedAF by 1.4%-12.9% and converges faster. According to Fig. 4(b), when LR and CNN models of MNIST dataset reach a training accuracy of 90%, the global communication time of CSA_FedVeh is 6.96 s and 3.808 s, respectively. This is a reduction of 16.6% and 19.83% compared to FedAF, a reduction of 62.07% and 49.56% compared to FedASY, a reduction of 47.76% and 37.31% compared to SAFA, and a reduction of 11.36% and 3.4% compared to R-SAFSA. According to Fig. 4(c), when LR and CNN models of GTSRD dataset reach a training accuracy of 75%, the global communication time of CSA_FedVeh was 15.6 s and 10.496 s, respectively. This is a decrease of 35.13% and 35.21% compared to FedAF, a decrease of 68.4% and 68.6% compared to FedASY, a decrease of 56.99% and 56.93% compared to SAFA, and a decrease of 1.01% and 7.54% compared to R-SAFSA. The results demonstrate that CSA_FedVeh achieves the optimal global communication time across all models. This implies that, with the same communication budget, CSA_FedVeh reduces communication costs between VUs and the ES by executing cluster-based semi-asynchronous aggregation.

In summary, our CSA_FedVeh framework has demonstrated superiority over four benchmarks in the following aspects. Firstly, as seen in Fig. 3, CSA_FedVeh consistently achieves better convergence than the benchmarks during training and reduces training time by approximately 24.6% to 60.2% to reach the same loss level. Secondly, as shown in Fig. 4(a), on both MNIST and GTSRD datasets, the average number of communication rounds for CSA_FedVeh is lower than

FedASY, SAFA, and R-SAFA, and only slightly more than FedAF. Additionally, CSA_FedVeh achieve the lowest total communication time, the highest accuracy and the fastest convergence speed. Finally, Figs. 4(b) and 4(c) demonstrate that, compared to the baselines, CSA_FedVeh reduces communication costs by 3.4% to 62.07% on MNIST dataset and by 1.01% to 68.6% on GTSRD dataset, while achieving similar accuracy.

6 Conclusion

In this paper, we propose CSA_FedVeh, a novel cluster-based semi-asynchronous FL framework for IoV. We aspire to enhance the effectiveness of FL in the dynamic and intricate scenarios of IoV. Under this guidance, We proposed the STW-DBSCAN clustering algorithm, which takes advantage of Non-IID of data collected by vehicles to cluster vehicles with similar vehicle space-time location and high model weights similarity, efficiently addressing the straggler problem and accelerating global model training. Meanwhile, we combine the semi-asynchronous federated aggregation mechanism to accelerate the speed of global aggregation. The experimental results indicate that our proposed framework can obtain excellent performance under resource constraints on the datasets of Non-IID compared with baselines. In the future, we will explore our CSA_FedVeh framework on vehicle tasks that require stable, low-latency, and highly reliable services in IoV, such as object tracking, high-definition (HD) map generation and augmented reality (AR) navigation.

Acknowledgment. This work was supported by the National Natural Science Foundation of China (No.62272063, No.62072056 and No.61902041), the Natural Science Foundation of Hunan Province (No.2022JJ30617 and No.2020JJ2029), open research fund of Key Lab of Broadband Wireless Communication and Sensor Network Technology, Nanjing University of Posts and Telecommunications (No.JZNY202102), Standardization Project of Transportation Department of Hunan Province (B202108), Hunan Provincial Key Research and Development Program (2022GK2019) and the Scientific Research Fund of Hunan Provincial Transportation Department (No.202143).

References

1. Banabilah, S., Aloqaily, M., Alsayed, E., Malik, N., Jararweh, Y.: Federated learning review: fundamentals, enabling technologies, and future applications. *Inf. Process. Manage.* **59**(6), 103061 (2022)
2. Chaudhry, S.A.: Designing an efficient and secure message exchange protocol for internet of vehicles. *Secur. Commun. Netw.* **2021**, 1–9 (2021). <https://doi.org/10.1155/2021/5554318>
3. Liu, S., Liu, L., Tang, J., Yu, B., Wang, Y., Shi, W.: Edge computing for autonomous driving: opportunities and challenges. *Proc. IEEE* **107**(8), 1697–1716 (2019). <https://doi.org/10.1109/jproc.2019.2915983>
4. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: *Artificial Intelligence and Statistics*, pp. 1273–1282. PMLR (2017)

5. Cao, H., et al.: Prevention of GAN-based privacy inferring attacks towards federated learning. In: Gao, H., Wang, X., Wei, W., Dagiuklas, T. (eds.) Collaborative Computing: Networking, Applications and Worksharing. CollaborateCom 2022. LNICS, Social Informatics and Telecommunications Engineering, vol. 461, pp. 39–54. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-24386-8_3
6. Pan, S.J., Yang, Q.: A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **22**(10), 1345–1359 (2009). <https://doi.org/10.1109/tkde.2009.191>
7. Li, Q., Diao, Y., Chen, Q., He, B.: Federated learning on non-iid data silos: an experimental study. In: 2022 IEEE 38th International Conference on Data Engineering (ICDE), pp. 965–978. IEEE (2022). <https://doi.org/10.1109/icde53745.2022.00077>
8. Huang, X., Li, P., Yu, R., Wu, Y., Xie, K., Xie, S.: Fedparking: a federated learning based parking space estimation with parked vehicle assisted edge computing. *IEEE Trans. Veh. Technol.* **70**(9), 9355–9368 (2021). <https://doi.org/10.1109/tvt.2021.3098170>
9. Liang, F., Yang, Q., Liu, R., Wang, J., Sato, K., Guo, J.: Semi-synchronous federated learning protocol with dynamic aggregation in internet of vehicles. *IEEE Trans. Veh. Technol.* **71**(5), 4677–4691 (2022). <https://doi.org/10.1109/tvt.2022.3148872>
10. Huang, J., Xu, C., Ji, Z., Xiao, S., Liu, T., Ma, N., Zhou, Q., et al.: AFLPC: an asynchronous federated learning privacy-preserving computing model applied to 5g-v2x. *Security and Communication Networks 2022* (2022). <https://doi.org/10.1155/2022/9334943>
11. Ma, M., Wu, L., Liu, W., Chen, N., Shao, Z., Yang, Y.: Data-aware hierarchical federated learning via task offloading. In: GLOBECOM 2022–2022 IEEE Global Communications Conference, pp. 1–6. IEEE (2022). <https://doi.org/10.1109/globecom48099.2022.10000924>
12. Briggs, C., Fan, Z., Andras, P.: Federated learning with hierarchical clustering of local updates to improve training on non-iid data. In: 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1–9. IEEE (2020). <https://doi.org/10.1109/ijcnn48605.2020.9207469>
13. Tan, Y., Long, G., Liu, L., Zhou, T., Lu, Q., Jiang, J., Zhang, C.: FedProto: federated prototype learning across heterogeneous clients. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 8432–8440 (2022). <https://doi.org/10.1609/aaai.v36i8.20819>
14. Xie, C., Koyejo, S., Gupta, I.: Asynchronous federated optimization. arXiv preprint [arXiv:1903.03934](https://arxiv.org/abs/1903.03934) (2019)
15. Vu, T.T., Ngo, D.T., Ngo, H.Q., Dao, M.N., Tran, N.H., Middleton, R.H.: User selection approaches to mitigate the straggler effect for federated learning on cell-free massive MIMO networks. arXiv preprint [arXiv:2009.02031](https://arxiv.org/abs/2009.02031) (2020)
16. Wu, W., He, L., Lin, W., Mao, R., Maple, C., Jarvis, S.: Safa: a semi-asynchronous protocol for fast federated learning with low overhead. *IEEE Trans. Comput.* **70**(5), 655–668 (2020). <https://doi.org/10.1109/tc.2020.2994391>
17. Sun, J., et al.: FedSEA: a semi-asynchronous federated learning framework for extremely heterogeneous devices. In: Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems, pp. 106–119 (2022). <https://doi.org/10.1145/3560905.3568538>
18. Ma, Q., Xu, Y., Xu, H., Jiang, Z., Huang, L., Huang, H.: FedSA: a semi-asynchronous federated learning mechanism in heterogeneous edge computing. *IEEE J. Sel. Areas Commun.* **39**(12), 3654–3672 (2021). <https://doi.org/10.1109/jsac.2021.3118435>

19. Xiao, H., Zhao, J., Pei, Q., Feng, J., Liu, L., Shi, W.: Vehicle selection and resource optimization for federated learning in vehicular edge computing. *IEEE Trans. Intell. Transp. Syst.* **23**(8), 11073–11087 (2021)
20. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: *kdd.* vol. 96, pp. 226–231 (1996)
21. MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, pp. 281–297. Oakland, CA, USA (1967)
22. Hosmer Jr, D.W., Lemeshow, S., Sturdivant, R.X.: *Applied Logistic Regression*, vol. 398. Wiley, Hoboken (2013)
23. Shalev-Shwartz, S., Ben-David, S.: *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, Cambridge (2014)
24. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proc. IEEE* **86**(11), 2278–2324 (1998). <https://doi.org/10.1109/5.726791>
25. Stallkamp, J., Schlipsing, M., Salmen, J., Igel, C.: Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Netw.* **32**, 323–332 (2012). <https://doi.org/10.1016/j.neunet.2012.02.016>
26. Marfoq, O., Neglia, G., Bellet, A., Kameni, L., Vidal, R.: Federated multi-task learning under a mixture of distributions. *Adv. Neural. Inf. Process. Syst.* **34**, 15434–15447 (2021)