FedVANET: Efficient Federated Learning with Non-IID Data for Vehicular Ad Hoc Networks

Beibei Li[§], Yukun Jiang[§], Wenbin Sun[†], Weina Niu[‡], and Peiran Wang[§]

[§]School of Cyber Science and Engineering, Sichuan University, Chengdu, China 610065
[†]School of Electronics and Information, Northwestern Polytechnical University, Xi'an, China 710129
[‡]School of Computer Science & Engineering, University of Electronic Science & Technology of China, China
Email: {libeibei, jiangyukun, wangpeiran}@scu.edu.cn; sunwenbin@nwpu.edu.cn; vinusniu@gmail.com

***Abstract*—The vehicular ad hoc networks (VANETs) play a significant role in intelligent transportation systems (ITS). In recent years, federated learning (FL) has been widely used in VANETs to preserve the privacy-sensitive data, such as vehicle locations, drivers' driving patterns, on-board camera data, etc. However, conventional FL faces the challenges of non-independent and identically distributed (Non-IID) data and high communication overheads in VANETs. To address these challenges, we propose a novel FL framework for VANETs, named FedVANET, where a hierarchical inner-cluster FL model and a weighted inter-cluster cycling update algorithm are, respectively, developed. Extensive experiments demonstrate the high efficiency of the FedVANET in inner-cluster communications, effectiveness in handling Non-IID data, and robustness in dynamic VANET topologies.**

***Index Terms*—Vehicular ad hoc networks (VANETs), federated learning (FL), privacy preservation, Non-IID data.**

I. Introduction

The mobile ad hoc networks (MANETs) are centreless wireless networks composed of mobile nodes autonomously [1]. As specific MANETs for vehicles, the vehicular ad hoc networks (VANETs) play a significant role in intelligent transportation systems (ITS). In VANETs, there are two communication methods implemented to ensure real-time and useful information to drivers, which are respectively vehicle to infrastructure (V2I) and vehicle to vehicle (V2V) [2]. In the internet of vehicles (IoV), a large amount of users' data collected by global positioning system (GPS), camera, radar is used to improve the user experience. For instance, NVIDIA creates an end-to-end learning method that achieves accurate steering angle recognition using images and steering command captured by on-board sensors [3]. With the growing concerns about data privacy, recent years have witnessed increasing interest in applying federated learning (FL) into IoV [4] [5].

In FL, participants' data is stored locally, thus realizing data privacy preservation. When applying FL in VANETs, there are two major challenges: communication load and non-independent and identically distributed (Non-IID) data. In recent years, some multi-hop clustering algorithms for VANETs architectures have been proposed to effectively reduce the number of cluster heads (the vehicles in VANETs that directly communicate with base stations), as well as the clustering overhead [6]. Though these algorithms cannot diametrically cut down the communication load in FL, the proposed framework can lower the load based on these algorithms. The general multi-hop cluster VANET model is shown in Fig. 1. In this architecture, V2I and V2V communication utilize the universal mobile telecommunication system (UMTS) and the long term evolution (LTE) [7]. As for the Non-IID data, due to differences in regions and user groups, the distribution of collected data in different vehicles does not necessarily follow the law of independent and identical distribution (IID) [8]. FL inevitably faces the challenge of Non-IID data because it does not allow data to be separated from the local database. Experiments done by researchers show that the conventional FL with Non-IID data will greatly reduce the accuracy of the model compared to centralized learning [9], and a suitable mechanism is required to process Non-IID data. Though there are several state-of-the-art methods [10] [11] to deal with Non-IID data, our experiments show that they are not well adapted to VANETs. To overcome the above challenges, we design an efficient FL framework for VANETs. The contributions of our work are three-fold:

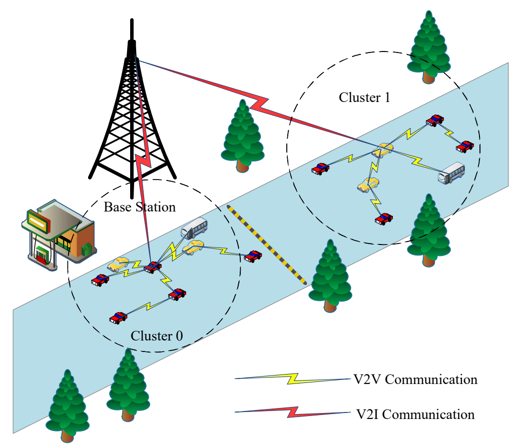


Fig. 1. The general multi-hop cluster VANET model.

- First, we propose an efficient FL framework that can be implemented in VANETs, which can support the general application of FL in VANETs.
- Second, an inner-cluster recursive mechanism is designed to reduce communication load and improve the model

performance. The recursive method ensures that each vehicle has the opportunity to participate in model training and reduces communication frequency between vehicles and the central server.
- Third, an inter-cluster cycling update algorithm is applied between central server and clusters to accelerate the convergence of the model and improve the robustness of the learning model with Non-IID data and dynamic inner-cluster topologies.

The remainder of this paper is organized as follows. In Section II and Section III we respectively elaborate on the goal of our framework and the proposed FedVANET. Then, in section IV, extensive experiments are designed and conducted to evaluate the efficiency and robustness of our proposed framework. Finally, we draw our conclusions in Section V.

II. Problem Formulation

In this section, we formulate the target problem of FL model training in FedVANET and address some constraints.

In FedVANET, there are M vehicles expecting to participate in model training, and the local dataset of vehicle m can be expressed as $D^m_{vehicle} := \{(X^m_a, y^m_a)\}_{a=0}^{|D^m_{vehicle}|-1}$, where X^m_a is the a-th data sample in vehicle m, y^m_a is the corresponding label of X^m_a. The M vehicles are distributed among N clusters, and there is no intersection between clusters. In addition, the dataset owned by all vehicles in cluster i can be expressed as $D^i_{cluster}$, $D^i_{cluster} = \cup_{m \in i} D^m_{vehicle}$. In general, our goal is to solve a distributed optimization problem

$$\begin{aligned}\min_{w \in \mathbb{R}^d} F(w) &= \sum_{i=0}^{N-1} \frac{|D^i_{cluster}|}{|D|} f(w; D^i_{cluster}) \\ &= \sum_{m=0}^{M-1} \frac{|D^m|}{|D|} f(w; D^m_{vehicle}),\end{aligned} \tag{1}$$

$$f(w; D^m_{vehicle}) = \frac{1}{|D^m_{vehicle}|} \sum_{a=0}^{|D^m_{vehicle}|-1} \mathcal{L}(w; X^m_a, y^m_a), \tag{2}$$

where w represents the weight of the trained network model, and F is a function representing the empirical risk of the current model over the dataset D. f is used to measure the empirical risk of a model over a specific dataset, and $\mathcal{L}$ represents the selected loss function.

When optimizing the problem, we address some constraints that exist in the scenarios of VANETs and FL. We assume a multi-hop cluster VANET model as the network model of FedVANET, in which vehicles need to meet four constraints:
- First, each vehicle can communicate with at least one (at most not limited) other vehicles in the cluster, but only cluster heads can communicate with the base stations.
- Second, the link is symmetric, that is, vehicles at both ends of the same communication link can transmit information to each other.
- Third, each vehicle needs to have appropriate storage space and computing capabilities for storing and updating network models.

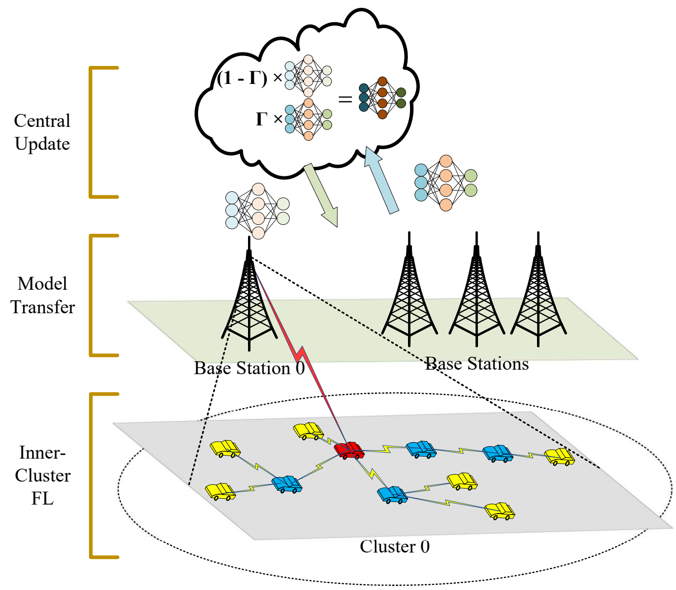


Fig. 2. The architecture of FedVANET.

- Fourth, the data collected by all vehicles is only stored locally to ensure data privacy.

III. FedVANET: Federated Learning Framework for Vehicular Ad Hoc Networks

In this section, we elaborate on the proposed FedVANET framework by expatiating Inter-Cluster and Inner-Cluster FL.

A. Inter-Cluster Federated Learning

Figure 2 shows the architecture of FedVANET, which consists of clusters containing several vehicles, base stations connecting to certain cluster heads, and a central server composed of a cycling decision unit (CDU) and a model update unit (MUU). In the Inter-Cluster FL, the central server coordinates clusters to participate in the model updating asynchronously. The coordination mechanism utilizes a novel weighted inter-cluster cycling update algorithm (shown in algorithm 1). In detail, its workflow can be described in four phases:

1) Training Initialization: The central server randomly generates the initialization model $W_{initial}$, then records the collection of clusters participating in model training in FedVANET. Besides, it is necessary to collect the distribution of the quantity of data contained in each cluster. For a specific cluster i, we consider its dataset and data quantity to be $D^i_{cluster}$ and $|D^i_{cluster}|$, respectively.

2) Training Cluster Selection: When round t starts, the CDU declares a set of clusters N_u, representing the set of clusters that have not participated in the update of the learning model in the current round. There are a total of K model updates in round t, and the central server sends the latest model parameter W_{t-} to a cluster selected by CDU in N_u each time the new update starts. The selection of clusters can be a fixed order or a random selection. Initially, W_{t-} is equal to $W_{initial}$, but it is assigned a value of $W_{(t-1)+}$ once the central has

Algorithm 1 Weighted Inter-Cluster Cycling Update

$W_{initial}$ $initialization$;
for each round $t = 0, 1, 2, ..., T-1$ **do**
  Initialize collection of clusters that need to participate model update $N_u = N$;
  if $t = 0$ **then**
    $W_{t-} = W_{initial}$;
  else
    $W_{t-} = W_{t-1}$;
  end if
  for each update $k = 0, 1, 2, ..., K-1$ **do**
    if $k = 0$ **then**
      $W^-_{t,k} = W_{t-}$;
    else
      $W^-_{t,k} = W_{t,k-1}$;
    end if
    Select cluster $i \in N_u$;
    Send $W^-_{t,k}$ to cluster i;
    $W^+_{t,k}$ = **Inner-Cluster**($W^-_{t,k}$);
    $W_{t,k} = F_{weight}(W^-_{t,k}, W^+_{t,k})$;
    Update $N_u = N_u - i$;
  end for
  $W_t = W_{t,K-1}$;
end for
Obtain the final model W_{T-1};

updated. After each update, the CDU removes the cluster that participated in the last model update from N_u.

3) $Inner-Cluster\ Training$: At the beginning of the k-th update in round t, the CDU sends the phased initialized model $W^-_{t,k}$ to a selected cluster in N_u for this cluster to perform Inner-Cluster training to obtain the model $W^+_{t,k}$. Inner-Cluster training is not visible to the central server, and its details will be introduced in the following subsection.

4) $Central\ Model\ Updating$: Whenever a cluster completes the Inner-Cluster training, the cluster will upload the trained model to the MDU for an update. For cluster i in round t, k-th update, we symbolize the model that the central server sends to and receives from the cluster as $W^-_{t,k}$ and $W^-_{t,k}$ respectively. The new model is updated by

$$F_{weight}(W^-_{t,k}, W^+_{t,k}) = (1-\Gamma)W^-_{t,k} + \Gamma W^+_{t,k} \tag{3}$$

$$\Gamma = b\frac{|D^i_{cluster}|}{|D^{average}_{cluster}|} = b\frac{|D^i_{cluster}|}{\sum_{i=0}^{N-1}|D^i_{cluster}|}, \tag{4}$$

where b represents a customized index and $|D^{average}_{cluster}|$ represents the average number of data samples of all clusters:

After generating the updated model $W_{t,k}$ with function F_{weight} in round t, k-th update, the cluster i needs to be eliminated from the collection N_u. Note that if k is equal to $K-1$, the trained model of round t will be set to $W_{t,K-1}$.

After T rounds (an empirically determined threshold) of FL model training, a satisfactory network model W_{T-1} will be obtained. It can be known from Algorithm 1 that each training round of FedVANET requires K times of Inner-Cluster training and K times of central model update. For conventional FL, the numbers are K and 1, but these additional calculations are negligible for the MDU.

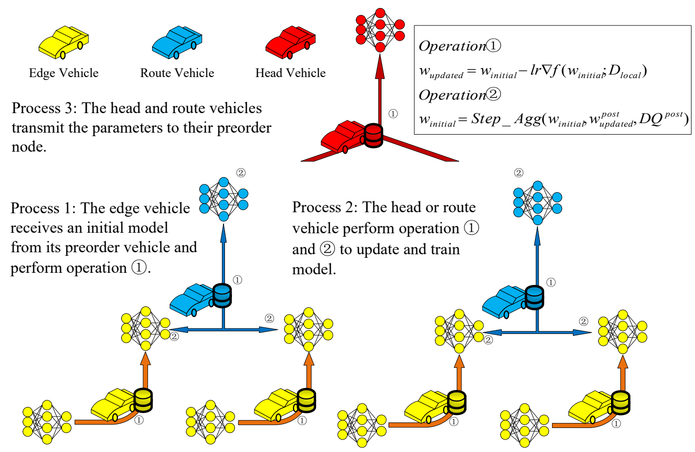


Fig. 3. The workflow of Inner-Cluster federated learning.

B. Inner-Cluster Federated Learning

Figure 3 illustrates the workflow of Inner-Cluster FL, the vehicles in FedVANET can be divided into three categories: head vehicle (v_{head}), route vehicle (v_{route}), and edge vehicle (v_{edge}). The detailed definition and complete workflow of these three types of vehicles are described as follows:

1) $Head\ Vehicle$: The v_{head} represents the only vehicle in a cluster that can communicate with base stations. It has no preorder vehicle (or the base station is its preorder vehicle), only postorder vehicles. When Inner-Cluster FL starts, it receives the model w^-_{head} that needs to be trained from the CDU and sends it to one of its postorder vehicles. We use $Post_{head}$ to denote the collection of all postorder vehicles of v_{head}. When a vehicle $m \in Post_{head}$ completes training, the trained model w_m and DQ^m (representing the data quantity used to generate the model w_m) will be sent to the v_{head} for step aggregation

$$Step\_Agg(w^-_{head}, w_m, DQ^m) = p_m w_m + (1-p_m)w^-_{head}, \tag{5}$$

where p_m denotes the proportion of DQ^m samples in all dataset in cluster i that

$$p_m = \frac{DQ^m}{|D^i_{cluster}|}. \tag{6}$$

Then, v_{head} sends the aggregated model w^-_{head} to one of the other postorder vehicles. The above steps are repeated until all $m \in Post_{head}$ have participated in the training, and finally, the v_{head} will perform model training with its local data $D^{head}_{vehicle}$

$$w_{head} = w^-_{head} - l_r\nabla f(w_{head}; D^{d_{head}}_{vehicle}), \tag{7}$$

to generate w_{head} and send it to the base station and MUU.

2) $Route\ Vehicle$: v_{route} symbolizes the vehicle that can play a routing role, which means that the v_{route} not only connects to a preorder vehicle, but also connects to at least one postorder vehicle. In Inner-Cluster FL, the v_{route} receives the learning model w^-_{route} from its preorder vehicle and transmits it to vehicles in its postorder vehicles collection $Post_{route}$ one by one. Similar to v_{head}, whenever the v_{route} receives

TABLE I
DATA DISTRIBUTION PATTERNS IN CLUSTERS.

Data Distribution	Cluster 0	Cluster 1	$\cdots$	Cluster 9
IID	$\overbrace{(0,1,\cdots,9),\cdots}^{10\ Vehicles}$	$\overbrace{(0,1,\cdots,9),\cdots}^{10\ Vehicles}$	$\cdots$	$\overbrace{(0,1,\cdots,9),\cdots}^{10\ Vehicles}$
Cluster-Level (L_C) Non-IID	$\overbrace{(0,\cdots,0),\cdots}^{10\ Vehicles}$	$\overbrace{(1,\cdots,1),\cdots}^{10\ Vehicles}$	$\cdots$	$\overbrace{(9,\cdots,9),\cdots}^{10\ Vehicles}$
Semi-Vehicle-Level (L_S) Non-IID	$\overbrace{(0,\cdots,0)}^{Vehicle\ 0},\overbrace{(1,\cdots,1)}^{Vehicle\ 1},\cdots,\overbrace{(9,\cdots,9)}^{Vehicle\ 9}$	$\overbrace{(0,\cdots,0)}^{Vehicle\ 0},\overbrace{(1,\cdots,1)}^{Vehicle\ 1},\cdots,\overbrace{(9,\cdots,9)}^{Vehicle\ 9}$	$\cdots$	$\overbrace{(0,\cdots,0)}^{Vehicle\ 0},\overbrace{(1,\cdots,1)}^{Vehicle\ 1},\cdots,\overbrace{(9,\cdots,9)}^{Vehicle\ 9}$
Fully-Vehicle-Level (L_F) Non-IID	$\overbrace{(0,\cdots,0),\cdots,}^{Vehicle\ 0\ to\ 4}\overbrace{(1,\cdots,1),\cdots}^{Vehicle\ 5\ to\ 9}$	$\overbrace{(1,\cdots,1),\cdots,}^{Vehicle\ 0\ to\ 4}\overbrace{(2,\cdots,2),\cdots}^{Vehicle\ 5\ to\ 9}$	$\cdots$	$\overbrace{(9,\cdots,9),\cdots,}^{Vehicle\ 0\ to\ 4}\overbrace{(0,\cdots,0),\cdots}^{Vehicle\ 5\ to\ 9}$

parameters returned by $m \in Post_{route}$, it updates w^{-}_{route} with function $Step\_Agg$ and sends it to the next selected postorder vehicle until all vehicles in collection $Post_{route}$ have participated in the FL. Then, the w_{route} and DQ^{route} are obtained through the calculation in v_{route} with its local data and sent to its preorder vehicle, the calculation on DQ^{route} is

$$DQ^{route} = |D^{v_{route}}_{vehicle}| + \sum_{m \in Post_{route}} DQ^{m}. \tag{8}$$

3) Edge Vehicle: v_{edge} refers to the vehicle in a multi-hop cluster VANET that only connects to a preorder vehicle but not any postorder vehicle. When the v_{edge} receives a learning model w^{-}_{edge} from its preorder vehicle, it immediately updates w^{-}_{edge} to acquire w_{edge}, and returns w_{edge} and DQ_{edge} to its preorder node, where $DQ^{edge} = |D^{v_{edge}}_{vehicle}|$.

When v_{head} completes model transmission, aggregation, and training, a phased updated FL model will be transmitted to the base station then MUU. Model training is performed once for each vehicle in a cluster, which is consistent with conventional FL. For a cluster with x vehicles, there are a total of x - 1 V2V links and 1 V2I link, and each link carries two parameters transmission in different directions during an Inner-Cluster training, which greatly reduces the communication frequency compared to the existing frameworks. In addition, though function $Step\_Agg$ is called x - 1 times by v_{head} and v_{route}, the calculation overhead is relatively ignorable.

IV. PERFORMANCE EVALUATION

In this section, extensive experiments are conducted to evaluate the performance of our proposed FedVANET framework.

A. Experiment Settings

1) Environmental Setup: The random Inner-Cluster topologies are generated by NetworkX [1] and the proposed FedVANET framework is implemented using Torch [2].

2) Clustering: We divide the vehicles into ten clusters, and each cluster contains ten vehicles. Each cluster has a cluster head connected to a base station to communicate with the central server, but the network topology in the cluster is self-organized. To simulate the shifty network topology of the multi-hop cluster VANET, we randomly generated ten sets of network topologies (see Fig. 4, the maximum hop count is not limited in our evaluation and the No. 0 node represents the cluster head), each containing ten nodes.

[1]A Python package for network structure (https:/networkx.org/)
[2]A Python deep learning library (https://pytorch.org/)

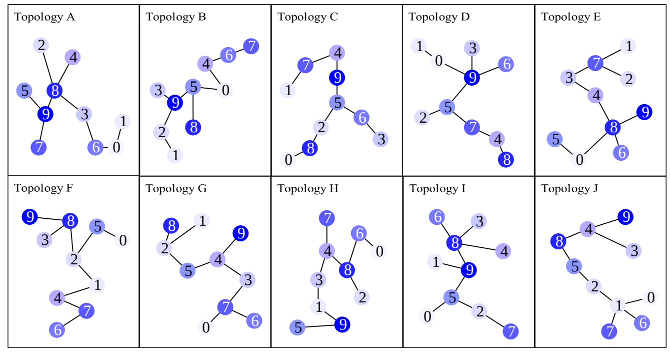
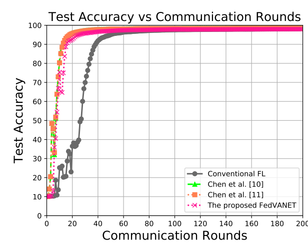
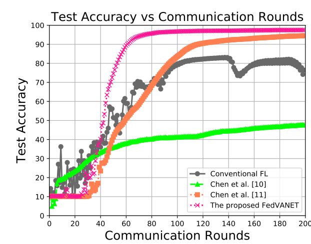


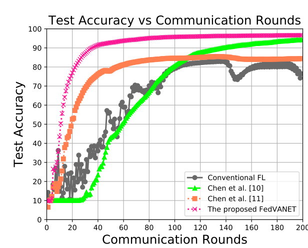
Fig. 4. Ten random topologies generated by NetworkX.



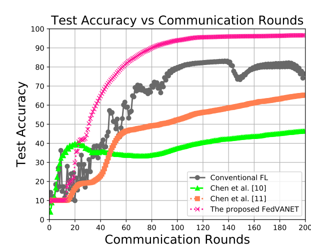
(a) IID



(b) L_C Non-IID



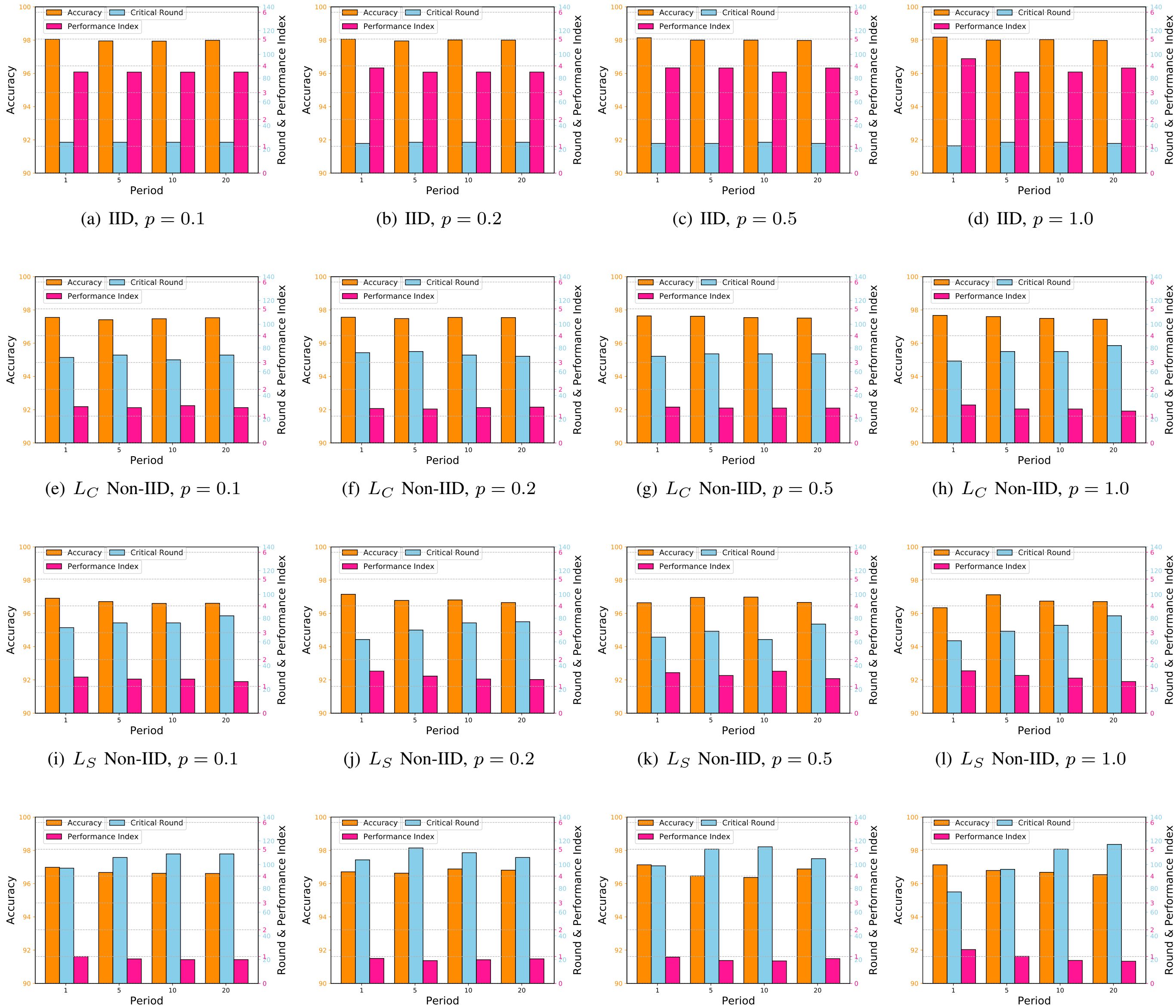
(c) L_S Non-IID



(d) L_F Non-IID

Fig. 5. Test accuracy figures for all schemes in four types of data distributions.

3) Dataset and Learning Model: To validate the effectiveness of our framework, like [10] and [11], the image dataset MNIST [12] including ten categories are selected for evaluation. Our experiments consider four types of data distributions (see TABLE I). While the MNIST dataset contains 60,000 training and 10,000 testing images, the number of training samples in each category is not equal. For Non-IID data distribution, we select 50,000 training samples from the



(a) IID, $p = 0.1$ (b) IID, $p = 0.2$ (c) IID, $p = 0.5$ (d) IID, $p = 1.0$

(e) L_C Non-IID, $p = 0.1$ (f) L_C Non-IID, $p = 0.2$ (g) L_C Non-IID, $p = 0.5$ (h) L_C Non-IID, $p = 1.0$

(i) L_S Non-IID, $p = 0.1$ (j) L_S Non-IID, $p = 0.2$ (k) L_S Non-IID, $p = 0.5$ (l) L_S Non-IID, $p = 1.0$

(m) L_F Non-IID, $p = 0.1$ (n) L_F Non-IID, $p = 0.2$ (o) L_F Non-IID, $p = 0.5$ (p) L_F Non-IID, $p = 1.0$

Fig. 6. Performance metrics of proposed FedVANET under four datadistributons and multiple dynamic inner-cluster topologies.

original 60,000 training samples, each category corresponding to 10,000 samples. Then we divide these 50,000 samples into 100 parts, each part contains 500 samples with the same label, and each vehicle is assigned to a specific part of the sample as its local data. For IID data distribution, 500 samples containing all ten categories are allocated to each vehicle. While the machine learning networks that could be used in this work are various, we conduct our experiments on LeNet-5 with stochastic gradient descent (SGD) optimizer, lr = 0.001, epoch = 2, batch size = 20, and total rounds = 200.

4) Baseline Studies: In this work, we compare the performance of our proposed FedVANET framework with conventional FL and some state-of-the-art studies in FL. Chen *et al.* [10] grouped the clients into several clusters and designed a sequential training manner to enable FL models sharing between the neighboring clients. Chen *et al.* [11] also conducted clustering and divided each learning round into multiple cycles of meta-update to boost the overall convergence. We fully reproduce these FL schemes in our experiments and compare them with our designed framework.

5) Performance Metrics: We take three metrics into consideration, namely accuracy, critical round, and performance index. The accuracy indicates the proportion of correct predictions made by the learning model with testing samples, and the critical round represents the number of training rounds when the model reaches a certain accuracy (95 % for MNIST). To better measure the performance of the model, we introduced the performance index, which is numerically equal to 100*accuracy divided by critical round.

B. Comparison with State-of-the-Art Studies

The curves of test accuracy for our proposed framework and other baselines are shown in Fig. 5, where all four types of data distributions are considered. We assign the generated ten sets of random inner-cluster topologies to ten different clusters for better comparison and assume that the topology remains unchanged in each cluster during training. Besides, the selection of clusters in Chen $et\ al.$ [11] and our proposed FedVANET follows a fixed order from cluster 0 to 9.

First, as demonstrated in Fig. 5(a), with IID data, while all schemes reach an accuracy rate of about 98.00%, the other three models reach similar faster model convergence rates than conventional FL's.

Second, conventional FL remains the test accuracy of 76.87% in terms of three Non-IID distributions, and our work reaches the highest accuracy that are 97.52%, 96.65% and 96.59%, respectively. Figure 5(b) shows accuracy with L_C Non-IID data, Chen $et\ al.$ [11] receives a relatively high accuracy (94.52%), but the accuracy for Chen $et\ al.$ [10] is only 47.58%. In Fig. 5(c), with L_S Non-IID data, Chen $et\ al.$ (94.27%) [10] and Chen $et\ al.$ [11] (84.38%) perform better than conventional FL but are inferior to our work. Unfortunately, Figure 5(d) shows that with L_F Non-IID data, the accuracy for Chen $et\ al.$ [10] and Chen $et\ al.$ [11] are 65.23% and 46.39%, even lower than that of conventional FL.

Overall, the experimental results in Fig. 5 indicate that compared with other schemes, our framework is able to achieve higher accuracy at the same communication round, and needs fewer communication rounds to reach the same accuracy. In a word, our work is more efficient than the existing schemes with higher accuracy and fewer communication rounds.

C. Robustness for Dynamic Inner-Cluster Topologies

Figure 6 compares the performance metrics of our proposed FedVANET under four data distributions and multiple dynamic topologies. p represents the proportion of clusters in all ten clusters whose inner-cluster topology is dynamic during FL and the period indicates how many rounds of training the topology in clusters will change once. Each changing topology is randomly selected from the generated ten topologies.

Figure 6(a) - 6(d) illustrate that with IID data, the proposed FedVANET can reach an accuracy of more than 97.94%, the critical round of smaller than 26, and the performance index of approximately 3.84. Besides, as shown in Fig. 6(e) - 6(p), under dynamic topologies of different levels, our work can obtain the accuracy, critical round, performance index of around 97.50%, 75, 1.30 when Non-IIDness is L_C, 96.80%, 72, 1.34 when Non-IIDness is L_S, and 96.70%, 99, 0.98 when Non-IIDness is L_F, respectively.

All in all, numerical results present high robustness of our proposed FedVANET under different data distributions when inner-cluster topology changes. Furthermore, as p rises, the performance index rises slightly, and as the period increases, it shows a slight downward trend.

V. Conclusion

In this paper, we have proposed a novel efficient FL framework with Non-IID data for VANETs, named FedVANET, which achieved efficient FL in VANETs and robustness in complicated dynamic topology scenarios. More specifically, As for inner-cluster vehicles, we created a recursive Inner-Cluster FL to improve communication-efficiency. Then, a weighted inter-cluster cycling update algorithm between the central server and clusters is proposed to enable the robustness of our framework under various data distributions and network topologies. Extensive experiments demonstrated that, compared with other schemes, the proposed FedVANET outperformed in terms of accuracy and communication-efficiency. Also, they exhibited the high robustness of our work with Non-IID data in dynamic topologies.

Our future work will focus on more specific vehicle-related datasets to further verify and tune our FedVANET framework.

References

[1] G. A. Walikar and R. C. Biradar, "A survey on hybrid routing mechanisms in mobile ad hoc networks," *J. Netw. Comput. Appl.*, vol. 77, pp. 48–63, Jan. 2017.

[2] K. Liu, J. K. Y. Ng, V. C. Lee, S. H. Son, and I. Stojmenovic, "Cooperative data scheduling in hybrid vehicular ad hoc networks: Vanet as a software defined network," *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1759–1773, Jun. 2016.

[3] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, Apr. 2016.

[4] S. Abdulrahman, H. Tout, H. Ould-Slimane, A. Mourad, C. Talhi, and M. Guizani, "A survey on federated learning: The journey from centralized to distributed on-site learning and beyond," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5476–5497, Apr. 2021.

[5] Q. Kong, F. Yin, R. Lu, B. Li, X. Wang, S. Cui, and P. Zhang, "Privacy-preserving aggregation for federated learning-based navigation in vehicular fog," *IEEE Trans. Ind. Informat.*, dOI: 10.1109/TII.2021.3075683.

[6] D. Zhang, H. Ge, T. Zhang, Y.-Y. Cui, X. Liu, and G. Mao, "New multi-hop clustering algorithm for vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 4, pp. 1517–1530, Apr. 2018.

[7] N. Prasad, M. Arslan, and S. Rangarajan, "Exploiting cell dormancy and load balancing in lte hetnets: Optimizing the proportional fairness utility," *IEEE Trans. Commun.*, vol. 62, no. 10, pp. 3706–3722, Oct. 2014.

[8] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, Jan. 2019.

[9] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *International Conference on Learning Representations (ICLR)*, Apr. 2020.

[10] Z. Chen, D. Li, M. Zhao, S. Zhang, and J. Zhu, "Semi-federated learning," in *IEEE Wireless Communications and Networking Conference (WCNC)*, May 2020, pp. 1–6.

[11] C. Chen, Z. Chen, Y. Zhou, and B. Kailkhura, "Fedcluster: Boosting the convergence of federated learning via cluster-cycling," in *IEEE International Conference on Big Data (Big Data)*, Dec. 2020, pp. 5017–5026.

[12] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.