Driving Towards Efficiency: Adaptive Resource-aware Clustered Federated Learning in Vehicular Networks
Khalil, Ahmad; Lotfian Delouee, Majid; Degeler, Victoria; Meuser, Tobias; Fernandez Anta, Antonio; Koldehofe, Boris

Published in:
The 22nd Mediterranean Communication and Computer Networking Conference (MedComNet’24)

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Final author's version (accepted by publisher, after peer review)

Publication date:
2024

Link to publication in University of Groningen/UMCG research database

Citation for published version (APA):

Copyright
Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license. More information can be found on the University of Groningen website: https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment.

Take-down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): http://www.rug.nl/research/portal. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.
Driving Towards Efficiency: Adaptive Resource-aware Clustered Federated Learning in Vehicular Networks

Ahmad Khalil, Majid Lotfian Delouee, Victoria Degeler, Tobias Meuser, Antonio Fernandez Anta, Boris Koldehofe

ABSTRACT— Guaranteeing precise perception for fully autonomous driving in diverse driving conditions requires continuous improvement and training. In vehicular networks, federated learning (FL) facilitates this by enabling model training without sharing raw sensory data. As an extension, clustered FL reduces communication overhead and aligns well with the dynamic nature of these networks. However, current literature on this topic does not consider critical dimensions of FL, including (1) the correlation between perception performance and the networking overhead, (2) the limited vehicle storage, (3) the need for training with freshly captured data, and (4) the impact of non-IID data and varying traffic densities. To fill these research gaps, we introduce AR-CFL, an Adaptive Resource-aware Clustered Federated Learning framework. AR-CFL utilizes clustered FL to collectively model the environment of connected vehicles, integrating models from all vehicles and ensuring universal accessibility to the refined model. AR-CFL dynamically enhances system efficiency by adaptively adjusting the number of clusters and specific in-cluster participant selection strategies. Using AR-CFL, we systematically study the scenario of online car detection model training on non-IID data across varied conditions. The evaluation results highlight the robust detection performance exhibited by the trained model employing the clustered FL approach, despite the constraints posed by limited vehicle storage capacity. Furthermore, our investigation unveils superior training performance with clustered FL in comparison to specific classical FL scenarios, increasing the training efficiency in terms of participating nodes by up to 25% and reducing cellular communication by 33%.

Index Terms—Vehicular Networks, Clustered Federated Learning, Adaptivity, Vehicular Perception, Deep Learning.

I. INTRODUCTION

Autonomous driving comes with the promise of making vehicles’ movements more predictable and less reliant on the drivers’ decisions, hence increasing road safety and traffic efficiency. However, today’s vehicles have a narrow perception of the environment due to limited onboard sensing [1]. To cope with this, exchanging collected data among vehicles (and all road users) can help achieve a better perception of the environment [2]. The evolving Vehicle-to-everything (V2X) technologies provide a means of communication between road users and enable them to collect and aggregate perception data cooperatively, in the so-called Collective Perception (CP) [3].

Deep Neural Networks (DNNs) have a pivotal role in individual perception and object detection in autonomous driving. These networks usually undergo centralized training before deployment, utilizing data that is limited in its coverage. Consequently, Deep Neural Network (DNN) models trained on such data may exhibit low performance in object detection, even with quality management techniques in place [4], [5]. To overcome this limitation, continuous online model training can be leveraged to enhance adaptability and ensure robust object detection performance for fully autonomous driving across diverse conditions [2].

Federated Learning (FL) is a technique to train DNN models from distributed data sources (e.g., using the computational resources of each road user). With FL, a central server maintains DNN models that are updated with the incremental changes in its parameters provided by the participants/clients. Since the data sent to the central server is usually much smaller than the raw training data, FL reduces communication requirements. Additionally, it protects (up to some level) the data owners’ privacy, which is important in many Internet of Things (IoT) applications (e.g., [6], [7]), by keeping the raw data stored locally.

Recent studies demonstrate that FL techniques can train DNNs with optimized client-server interactions [8]. Moreover, FL is a good paradigm for implementing collective perception techniques among heterogeneous resources with Non-Independent and Identically Distributed (non-IID) data (i.e., when the data from different parties has different characteristics). With these benefits in mind, continually training a DNN employing FL will solve some of the challenges of autonomous driving.

Nevertheless, the communication requirements between participants and the central server in FL may still be too high in certain conditions. Hence, clustering can be used to alleviate communication overhead. Clustered FL works by grouping clients nearby into a cluster, which requires extensive coordination and synchronization between the involved entities. But, it reduces the communication re-
requirements between clients and the central server, since most of the training data exchange occurs inside clusters [9]. Moreover, it can help enhance the performance of the FL approach by decreasing the time required to train the model with a considerable level of perception capability.

To leverage the benefits of Clustered FL at its fullest, we present an Adaptive Resource-aware Clustered Federated Learning framework, referred to as AR-CFL, specifically designed to comprehensively explore and optimize factors impacting online learning and communication needs within vehicular environments. Our innovative framework incorporates adaptive mechanisms to optimize system efficiency dynamically. Additionally, leveraging AR-CFL, we investigate the training of a DNN vehicle detection model on non-IID data under diverse conditions. We systematically compare and discuss the outcomes obtained under different design decisions and configuration options.

In summary, this paper’s contributions are:

1) A novel framework (AR-CFL) that extends the capabilities of FL with adaptive clustering to provide hierarchical FL (improving existing Clustered FL solutions). This leads to boosting environment perception capability (i.e., detection performance), reducing the volume of the exchanged data, and providing a fast-converging training process.

2) A novel Dynamic Sampling concept, introduced to more realistically consider the storage limitation of vehicles in V2X networks.

3) A new Dynamic Cluster Members Involvement strategy to dynamically adjust the number of clients participating in the learning process in each cluster.

4) Three new synthetic datasets, generated by employing the CARLA simulator, constitute a comprehensive evaluation benchmark.

5) The evaluation of AR-CFL with these datasets, provides interesting results and conclusions.

The rest of the paper is organized as follows. The related work is presented in Section II. We further detail the problem, particularly for a vehicular network scenario, and motivate the need for a mechanism to exploit the benefits of clustering in FL-based data analysis in Section III. Then, Section IV presents the details of AR-CFL. The evaluation results of AR-CFL are exhibited in Section V. Finally, Section VI concludes the paper and presents future work.

II. RELATED WORK

In this section, we review the literature in two key areas, Object Detection and Clustering, both using FL in vehicular context, identifying the research gaps.

A. FL-based Object Detection in Vehicular Context

How FL affects vehicular environments has been extensively explored in the literature showing comparable performance with traditional centralized learning while preserving user’s privacy [10]. The benefits can vary from improving the detection performance [11] to training models over heterogeneous data sources (i.e., vehicle’s onboard sensors) [12] by dynamically adjusting local training iterations and using model compression to reduce communication overhead during model exchanges [13]. However, resource allocation remains challenging and can be alleviated by optimizing network management with a multi-layer graph (e.g., [14]). On the other hand, trading some privacy for higher utility can be achieved by opting for clients with sufficient resources, while others send their datasets to the central server [15]. Even with FL, privacy still might be violated in the model exchange or client selection phases. However, initial solutions such as multi-layer context-aware client selection and aggregation [16] degrade privacy violations.

B. Clustered Federated Learning in Vehicular Context

Adapting to the dynamic client diversity in different vehicular network topologies can be achieved by a weighted inter-cluster cycling update algorithm [17]. In addition, imbalanced and distribution-shifted training data was handled by a flexible Clustered Federated Learning (CFL) framework that groups clients based on optimization direction similarities to reduce training divergence [18]. Cluster formation in FL can be done based on client data distribution, e.g., by incorporating game theory principles [19] or considering the benefits of platooning [20]. Also, CFL-based object detection techniques in vehicular networks have been studied briefly by utilizing Vehicular-to-Vehicular (V2V) resources to bypass the communication bottleneck (cf., [21]). Lack of adaptivity by the static formation of clusters (e.g., vehicles under a base station’s coverage) and the limitation of dealing with mobility (i.e., vehicle handover) to enable continuous training in similar research works indicate that further studies are required.

C. Research Gaps

Several noteworthy research gaps necessitate further investigation, outlined as follows:

- **Application-Communication Network Integration:** A notable gap exists in the field of CFL, particularly in examining the relationship between exchanged data volume, influencing communication overhead, and the associated impact on application-related performance.

- **Limited Storage Consideration:** Existing CFL techniques overlook the consideration of limited vehicle storage, which leads to unrealistic performance evaluation.

- **Training on Freshly Collected Data:** The imperative need for training models with freshly collected data over successive iterations rather than static datasets is insufficiently addressed in current CFL approaches.

- **Influence of Varied Traffic Densities:** The literature lacks exploration into the impact of varying traffic densities on online training systems within the context of CFL.

- **Dynamic Clustering Participation:** The evaluation of CFL approaches has not explored varying cluster counts and the involvement of various vehicles within clusters.

III. CASE STUDY: OBJECT DETECTION MODEL TRAINING IN VEHICULAR NETWORKS

We demonstrate how to benefit from involving clustering in designing a two-level FL approach in the scenario of online model training of object detection applicable to
autonomous driving. Although FL is proven capable of assisting object detection [2], the amount of exchanged data and the convergence time are considerably increased to achieve an acceptable detection capability level. Hence, achieving optimum values for both factors requires modifying the conventional FL mechanism. When a cluster of vehicles is selected as an FL client, a new set of learning rounds will be initiated within the cluster, called Intra-Cluster Federated Learning. The trained model is exchanged directly between cluster members using V2V communication. It experiences multiple rounds of training within the cluster before being shared with the server. This approach reduces data exchange via cellular communication. Additionally, stable direct V2V communication within the cluster decreases convergence time.

As depicted in Figure 1, the training involves three primary entities:

a) Road User: A vehicle or any other entity participating in the process that generates data streams about the surrounding environment is called a road user. Each road user has its own onboard sensing configurations (e.g., a camera or lidar sensor) [22]. By merging the list of detected objects with the spatio-temporal information, a road user can build its own local environment model that will be used in subsequent decision-making processes. Also, vehicles can exchange perception data, extending spatial awareness above their limited perception. Such perception data is encapsulated in collective perception messages [23] that can be exchanged directly between road users or through any intermediate node (e.g., an edge server) [24].

For the sake of simplicity, we assume that all road users have similar sensing deployment to generate the same data type and operate with identical environment models.

b) Edge Server: In a conventional scenario, an edge server acts as a simple base station that only forwards data. Besides communicating with road users, edge servers can also exchange data with each other [25], [26]. That is why our mechanism not only provides a collective perception for road users who are within the communication range of each other but also beyond such a spatial limitation.

c) Cloud Server: The principal functionality of this component is to orchestrate the whole process of collective perception in object detection. It is responsible for initiating the detection task that should be performed based on the road users’ collective perception. In this regard, the cloud server is working closely with the edge servers to distribute the detection models among them. In addition, it benefits from the results of the learning process performed in the edge servers by combining the learning parameters.

Each cluster can be formed by moving road users (e.g., a combination of moving vehicles) and by a set of non-moving users (e.g., a group of parked vehicles plus those waiting behind a traffic light). Notice that increasing the number of cluster members is a double-edged sword, increasing the computing capabilities and membership dynamicity. Each cluster has a Cluster Head (CH), that manages and coordinates the rest of the Cluster Members (CMs) and exchanges information with them via the WiFi network.

IV. AR-CFL SYSTEM DESIGN

In Figure 2, we illustrate the main components of AR-CFL. When a user issues a task to the central server, a global model is generated to be trained online. In our case study, a user could be a car manufacturer who aims to improve the perception models of his vehicles. For example, if the task aims to detect object 1 in Section III, the model concerns the dimensions and position of this object. Then, the global model will be pre-trained by a random sample of vehicles’ data. Next, the central server decides the required vehicles based on this specific task. Finally, the new global model will be generated by aggregating the model updates trained in the selected clients. The entire learning process in this step is called Inter-Cluster Federated Learning (RCFL), and it includes the processing flow from the central server (i.e., in the cloud layer) to the participants (i.e., vehicles in the edge layer). The second level of learning is within each cluster, called Intra-Cluster Federated Learning (ICFL). Here, the cluster head decides the required number of local iterations.
(i.e., training rounds within the cluster) according to the cluster’s available computing resources.

Each task requires a particular amount of data and computing resources to be executed. Moreover, Therefore, the Resource Manager module aims to adjust control variables $\beta$ (in-cluster participant selection strategy) and $\eta$ (number of in-cluster participants) to match the required Computing Resources (CR) for each task. These parameters collectively determine the total number of participants $N_n$. Please refer to Table I for a comprehensive list of the evaluation parameters utilized in our study.

A. RCFL Framework

In Algorithm 1, we present the procedure of client selection and synchronized online learning from the cloud server point of view. We chose Federate Averaging (FedAvg) [27], a pioneering aggregation approach that achieves better accuracy in previous studies [28], in which CR (i.e., $CR_{max}$) by utilizing a small set of data gathered from all potential participants. Moreover, the participant selection strategy within the cluster (i.e., $\beta$) is determined here by the central server and sent to the cluster heads so that all clusters perform the selection uniformly. Various selection strategies have been introduced in the proposed approach that will be elaborated on further. Also, the number of participants that should be chosen in the specified selection strategy is determined here using the $\eta$ value.

Unlike FedAvg which chooses random clients, this approach selects as many cluster participants as possible. To this end, it selects clients with the highest CR values (i.e., $CR_{max}$), as clusters usually have higher computing resources. It continues selecting clients until their computing summation (i.e., $R_g$) reaches the threshold required for the task. Then, each chosen participant receives the global model $\omega_g$ and replaces its current local model $\omega_g^c$.

Suppose the client is of a type of cluster. In that case, the second level of FL is started within the cluster (see Algorithm 2). In the end, the cluster head uploads the cluster’s aggregated update to the central server. Otherwise, if the client is a single vehicle, it partitions the local data into batches of size $B$ and repeatedly applies the model to these data blocks for $E$ number of iterations, e.g., using Stochastic Gradient Decent (SGD). This generates the updated local model $\omega_{g+1}$, which is uploaded to the central server. Finally, the received trained local models are aggregated in the central server using a weighted sum into the new global shared model $\omega_{g+1}$. Notice that the weight for each locally trained model is calculated based on the number of performed iterations for each client $c$ (i.e., $E_c$) over the total iterations in this global training round (i.e., $E_g$). This way, more importance is given to the cluster updates, which were trained with more local training iterations.

B. ICFL Framework

The second level of FL is illustrated in Algorithm 2. After initializing the cluster member set and the local training model, in each local training iteration, the cluster head opts for the set of participants according to the in-cluster participant selection strategy (i.e., $\beta$) and also the $\eta$ value. There are three selection strategies for selecting in-cluster participants $\beta$: Full Aggregation, Random, and MaxLabel. When $\beta = \text{Full Aggregation}$, all cluster members would participate, and models are aggregated at the cluster head. On the other hand, when $\beta = \text{Random}$, FL clients are randomly selected in each iteration. Finally, when $\beta = \text{MaxLabels}$, cluster members with the most labels (data-rich) are selected (See Figure 3).

In both Random, and MaxLabel strategies, $\eta$ specifies how many members will be involved in the local training within each cluster. The central server determines $\eta$ based on various conditions (e.g., task, vehicle’s computation capabilities, etc.). E.g., $\eta = 2$ means that two clients from each cluster will be selected to participate in this local iteration. When having $\eta = 2$ and the $\beta = \text{MaxLabels}$, two cluster members with the largest label count in that specific cluster are selected. In the case of FullAggregation setup, $\eta$ equals the total number of cluster members including CH. In the next step, each selected member trains the model with fresh data and returns its update to the cluster head. The collected updates will be aggregated into a new model and be used for the next local iteration in this cluster. Once the training rounds are finished, the cluster head returns the last aggregated update to the

---

**Algorithm 1** RCFL Procedure

1. **Initialization:**
   - $C \leftarrow \{c_1, CR_{c_1}, \ldots, c_n, CR_{c_n}\}$; $\triangleright$ clients
   - $CR_{task} \leftarrow$ Minimum Required Resources for task;
   - Pre-trained $\omega$; $\triangleright$ Initial model
   - $\beta \leftarrow$ In-Cluster Participant Selection Strategy;
   - $\eta \leftarrow$ Number of In-Cluster Participants;

2. **for** global iteration $g = 0, 1, \ldots$ **do**
   - **Update(C);** $\triangleright$ Sum of Computing Resources
   - $E_o \leftarrow 0$; $\triangleright$ Sum of Client’s Weights
   - $C_q \leftarrow \{\}$; $\triangleright$ Selected Clients
   - $C' \leftarrow \text{Sort (C, CR)}$; $\triangleright$ Descending Sorted Client Set
   - **while** $R_g \leq CR_{task}$ **do**
   - - $\sigma_{max} \leftarrow C'_{[0]}$; $\triangleright$ Client with the highest CR value
   - - $C_q \leftarrow C_q \cup \sigma_{max}$
   - - $R_g \leftarrow R_g + CR_{max}$
   - - $C' \leftarrow C' - C'_{[0]}$;
   - **end:**
   - **Distribute** $\omega_{g+1}$ to clients in $C_q$; $\triangleright$ Global model
   - **for** client $c \in C_q$ **do**
   - - $\omega^c_{g+1} \leftarrow$ FullAggregation $\omega_{g}^c$; $\eta$, $\beta$; $\triangleright$ Algorithm 2
   - **end:**
   - **end:**

3. **if** $c$ is a cluster **then**
   - $\omega^c_{g+1} \leftarrow$ FullAggregation $\omega_{g}^c$, $\beta$, $\eta$; $\triangleright$ Algorithm 2

4. **else**
   - Single client training
   - $P_t \leftarrow$ batch size of $B$;
   - Data Partitions
   - $E_c \leftarrow |P_t|$; $\triangleright$ Number of training rounds for $c$
   - **for** partition $p \in P_t$, do
   - - $\omega^c_{g+1} \leftarrow$ LocalTraining $\omega^c_{g}$, $\beta$, $\eta$;
   - **end:**
   - $E_o \leftarrow \sum_{c \in C} E_c$;
   - $\omega_{g+1} \leftarrow \sum_{c \in C} (E_c/E_o) \times \omega^c_{g+1}$; $\triangleright$ Trained model

Algorithm 2 ICFL Procedure

1: Initialization:
   \( \omega_0 \leftarrow \omega_{\text{Cluster}}^0 \) >> Initial model in cluster
2: for local iteration \( l = 0, 1, ..., k \) do
3:   \( C_l \leftarrow \emptyset \) >> Selected members for training
4:   if \( \beta = \text{Full}_\text{Aggregation} \) then
5:     \( C_l \leftarrow \text{Cluster} \);
6:   else if \( \beta = \text{Random} \) then
7:     for \( i \leftarrow 1 \) to \( \eta \) do
8:       \( c_i \leftarrow \text{Random(Cluster)} \);
9:     \( C_l \leftarrow \text{Cluster} - c_i \);
10:   else if \( \beta = \text{MaxLabel} \) then
11:     for \( i \leftarrow 1 \) to \( \eta \) do
12:       \( c_i \leftarrow \text{max}_{j \in \text{Cluster}} \text{LabelCount}_{m_j} \);
13:     \( C_l \leftarrow \text{Cluster} - c_i \);
14:   end if
15: end if
16: Distribute \( \omega_l^\text{Cluster} \) to clients in \( C_l \);
17: for \( c_i \in C_l \) do
18:   \( \omega_{l+1}^c \leftarrow \omega_{\text{Cluster}}^c \);
19:   \( \omega_{l+1}^\text{LocalTraining} \leftarrow \omega_{l+1}^c \);
20: \( \Omega \leftarrow \Omega \cup \omega_{l+1}^\text{Cluster} \); >> Set of Collected Updates
21: \( \omega_{l+1}^{\text{Cluster}} \leftarrow \text{Aggregate}(\Omega) \);
22: return \( \omega_k^{\text{Cluster}} \) >> Final in-cluster trained model

V. EVALUATION

In this section, we explore the scenario of online car detection model training using our AR-CFL framework. Our investigation focuses on two key aspects: online learning efficiency and communication overhead. The evaluation covers the following:

1) We analyze how varying traffic density influences the overall system performance.
2) We investigate how CFL enhances the efficiency of online learning in terms of communication overhead and learning efficiency, as compared to the centralized learning approach and to the classical FL approach.
3) We explore the influence of various selection strategies (i.e., \( \beta \)), the number of participants in the cluster (i.e., \( \eta \)), and the number of clusters (i.e., \( N_{\text{cls}} \)) on the overall performance.

Our evaluation considers several restrictions regarding vehicle equipment and data distribution:

1) Non-IID Data: We used non-IID data, with a clear characterization of data heterogeneity across system members.
2) Environmental Considerations: The evaluation is carried out under various environmental conditions to assess the robustness and adaptability of the CFL approach.
3) Communication Assumptions: We assume that communication between vehicles within the same cluster is easier to establish and less costly than communicating with edge or cloud servers (cellular communication).

A. Evaluation Scenario and Experimental Setup

We outline now the evaluation scenario and experimental setup aimed at assessing our approach. Our focus lies in training a car detection model using image data captured from the participating vehicles. To meet the requirement of having image data from multiple vehicles in similar conditions, we created a synthetic dataset using the CARLA simulator [29]. The experiments employed the YoloV8n model [30] as a car detection model. We use a Linux server with an NVIDIA RTX3090 Ti GPU for running the experiments. We list all evaluation parameters with their descriptions in Table I.

1) Considered Conditions: Our study involves data collection in various weather and lighting scenarios (as illustrated in Figure 4). These include clear weather daytime, rainy weather day-time, clear weather night-time, and rainy weather night-time. Experiments were conducted using a combination of these conditions.

2) General Setup Variables: The total vehicle count is set at \( (N_v = 12) \). We vary traffic density (\( \alpha \)) with values of 30, 50, and 100, where \( \alpha = 50 \) indicates the presence of 50 vehicles. These vehicles are distinct from the aforementioned 12 data collector vehicles participating in FL model training. Furthermore, we investigate various scenarios by adjusting the number of clusters \( (N_{\text{cls}}) \) to either 2 or 4, evenly distributing the clients across the clusters.

C. Handling The Limited Storage Challenge

Unlike the conventional FL approaches, we take into account the constrained storage capacity of the participating vehicles by purging the utilized data in each training round, making room for the acquisition of fresh data. This mechanism reflects a heightened degree of realism and effectiveness, particularly in the context of vehicular Federated Learning scenarios, where we factor in the inherent limitations of onboard storage and the dynamic real-time conditions surrounding the participating vehicles.

Besides, Dynamic Involved Members concept helps the central server to be flexible in the number of cluster members that would participate in each round of training. By combining parameters \( \beta \) and \( \eta \), clients of the second level of learning can be adjusted according to the learned insights from the previous iterations. For example, the size of the FL participant set can be decreased to save resources in case the detection performance is not improved by involving more cluster members.

Fig. 3: Example of two image samples. Here, the image in (b) is more data-rich than the image in (a).
TABLE I: Evaluation parameters with descriptions.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta$</td>
<td>In-cluster member selection strategy</td>
</tr>
<tr>
<td>$\eta$</td>
<td>Number of participants in cluster</td>
</tr>
<tr>
<td>$N_v$</td>
<td>Total number of vehicles</td>
</tr>
<tr>
<td>$N_p$</td>
<td>Number of participating vehicles</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Traffic density (30, 50, 100)</td>
</tr>
<tr>
<td>$N_{cl}$</td>
<td>Number of clusters</td>
</tr>
<tr>
<td>$E_g$</td>
<td>Number of global iterations (epochs)</td>
</tr>
<tr>
<td>$g$</td>
<td>Global iterations index</td>
</tr>
<tr>
<td>$E_l$</td>
<td>Number of local iterations</td>
</tr>
<tr>
<td>$b_s$</td>
<td>Batch size</td>
</tr>
<tr>
<td>$lr_0, lr_f$</td>
<td>Learning rate parameters</td>
</tr>
<tr>
<td>$tr$</td>
<td>Total training time</td>
</tr>
<tr>
<td>$e_d$</td>
<td>Total volume of exchanged data for training</td>
</tr>
<tr>
<td>$ed_l$</td>
<td>Data volume (cellular communication)</td>
</tr>
<tr>
<td>$ed_d$</td>
<td>Data volume (direct V2V communication)</td>
</tr>
</tbody>
</table>

Fig. 4: Examples of weather and lighting conditions considered in our study. The generated datasets are characterized by well-balanced distributions, ensuring that each condition constitutes approximately 25% of the total dataset samples.

3) Baselines: We benchmark our approach against two main baselines:

Centralized: Represents the optimal oracle scenario where all data collected from vehicles is centrally stored and used for model training.

ClassicalFL: In this case, no clustering is considered, and all FL clients (vehicles) are at the same level, and communicating directly with the central aggregation server using cellular communication.

4) Federated Learning Hyper-parameters: We chose the total number of global iterations as $E_g = 50$. Upon receiving the model, each client engaged in $E_l = 100$ local training iterations on the currently available chunk of the local data. The batch size was set to $b_s = 16$. We established the learning rate parameters with $lr_0$ and $lr_f$, configured at their default values of $lr_0 = lr_f = 0.01$. Additionally, we selected optimizer = auto while maintaining default values for all other model training and validation parameters [31].

B. Evaluation Metrics

To assess the different approaches, we examine them within two evaluation categories:

1) Learning Efficiency:

a) Detection performance: The two key metrics often used to evaluate the detection model’s performance are:

- mean Average Precision (mAP): This metric considers precision and recall across multiple object classes [32]. mAP is particularly valuable because it considers the object detection performance at different confidence score thresholds, making it a robust evaluation metric. In our study, we measure mAP50 (IoU $\geq 0.5$).
- F1 Score: We use the F1 score as a supportive metric to measure the trained model’s detection performance.

b) Training time: The total training time is denoted as $tr$ and measured in minutes. We omitted the model exchange time for the sake of simplification. Moreover, we excluded the selection time for participating clients. We considered the actual model training time and the model aggregation time.

2) Communication Overhead: We define $e_d$ to measure the size of the exchanged data while neglecting the generated traffic to select the participating clients in the CFL setups. In addition, we omitted all the other Collective Perception Message (CPM) loads for simplicity. In the case of Centralized setup, $e_d$ is calculated by measuring the size of the data samples (images) that are sent from the $N_v$ vehicles to the server, as follows:

$$e_d = \sum_{i=1}^{N_v} \sum_{j=1}^{N_e} k_i \cdot d_{as}(i,j)$$

where $k_i$ is the number of data chunks collected in vehicle $i$, and $d_{as}(i,j)$ is the data size $j$ from the vehicle $i$. On the other hand, for ClassicalFL setup, we exchange the models instead of raw data. The exchanged data volume here is relevant to the number of selected clients $N_p$ in each global iteration $g$. Upon finishing the training on the number of local iterations $E_l$, each selected vehicle sends the model back to the server. In this case, the final formula to calculate $e_d$ is as follows:

$$e_d = \sum_{g=1}^{E_g} 2 \times N_p \times model_s$$

where $model_s$ indicates the model size.

Finally, we consider two-level aggregation while computing data exchange volumes in the different CFL setups. This involves two communication types, cellular and direct V2V communication. Cellular communication is required between the server and cluster heads. By minimizing data exchange in this costly and delayed communication type, the overall system efficiency improves. The bandwidth cost, denoted as $ed_l$, is computed as

$$ed_l = \sum_{g=1}^{E_g} 2 \times N_{cl} \times model_s$$

replacing $N_p$ with $N_{cl}$. On the other hand, direct V2V communication is required between cluster heads and members and is faster and...

### D. Results and Discussion

1) **Analyzing Traffic Density Impact on Performance:**

We explore the influence of traffic density on the performance, considering three different values $\alpha = 30, 50, 100$. Figure 6 illustrates some detection performance values in the Centralized approach, revealing an evident trend: as traffic density rises, there’s more consistent model performance and an overall enhancement in the detection performance. Increased traffic density results in capturing more objects within the generated images, thereby enhancing the model training performance.

Notably, heightened traffic density correlates with increased detection performance without influencing training time or data exchange volume. These aspects depend solely on sample size and not on the characteristics within the samples. These observations go beyond the Centralized approach and are applicable across all other approaches.

2) **Influence of Clustering vs. Centralized Approach on Online Learning Efficiency:**

Examining the impact of CFL on online learning efficiency versus centralized learning, we emphasize the communication overhead and application-related performance. Figures 7 and 9 reveal that the Centralized approach consistently outperforms CFL setups in detection performance. Despite this, the gap remains constant across different $\alpha$ values.

The true advantage of both FL and CFL (i.e., clustering) emerges in reduced training time. As illustrated in Table II, both approaches demonstrate an impressive 52% decrease in training time compared to Centralized training approach. Furthermore, as illustrated in Figure 8, the CFL setups demonstrate a significant reduction of approximately 30% in the exchanged data volume compared to the Centralized approach when involving all clients in the training process.

3) **Clustering vs. ClassicalFL:**

We analyze how CFL impacts online learning efficiency compared to the classical federated learning approach.

Figures 10 and 11 offer nuanced insights into online learning efficiency. FullAggregation and MaxLabels CFL...
Fig. 7: Comparing the detection performance between the ClassicalFL approach and selected Clustering approaches under a traffic density of $\alpha = 30$.

Fig. 8: Comparison of exchanged data volume across different approaches relative to the number of participating clients $N_n$ and the number of clusters $N_{cls}$. All CFL setups exhibit consistent values across varying traffic densities.

strategies outperform traditional ClassicalFL in detection performance. FullAggregation involves all clients in each iteration, contrasting with ClassicalFL, which randomly selects a subset ($N_n$) of clients per iteration. However, FullAggregation introduces increased short-range communication ($e_d$) compared to ClassicalFL.

MaxLabels surpasses ClassicalFL by selecting clients within each cluster with the maximum labels per iteration, enhancing the detection performance and convergence. However, as illustrated in Figure 8, CFL introduces additional communication overhead. Nevertheless, with the increased number of participants, there is a decrease in $e_d$ and an increase in $e_{d_{cls}}$, highlighting the benefits of CFL over ClassicalFL. The Random CFL setup with $\eta = 1$ shows comparable performance to ClassicalFL, randomly selecting clients in each iteration. With increased

Fig. 9: Comparing the detection performance between the ClassicalFL approach and selected Clustering approaches under a traffic density of $\alpha = 100$.

Fig. 10: Comparing the detection performance between the ClassicalFL (Nn=2) approach and selected Clustering approaches under a traffic density of $\alpha = 100$ with one selected client at each cluster ($\eta = 1$).

$\eta$ (e.g., $\eta = 3$), Random CFL outperforms ClassicalFL in detection performance due to more participating clients.

4) Impact of In-cluster Member Selection Strategy & Varying Cluster Numbers on Overall CFL Performance:
We examine the influence of changing the total number of clusters ($N_{cls}$), diverse selection strategy ($\beta$), and involved cluster members ($\eta$) on overall system performance, using $\beta = \text{FullAggregation}$ as a baseline for comparison.

Varying Cluster Numbers ($N_{cls}$): As shown in Table II, the training time consistently stands at $t_{tr} = 101$ minutes across varying numbers of clusters $N_{cls}$. Examining the communication overhead illustrated in Figure 8, we observe an increase in long-range communication overhead corresponding to the increased values of $N_{cls}$. This can be attributed to the increased communication overhead between the head nodes of clusters and the server. Similarly, the short-range communication overhead exhibits a rising trend with an increased number of clusters. This trend indicates a broader engagement of cluster nodes in online learning.

Turning our attention to detection performance, as depicted in Figure 12, we found that when $\beta = \text{FullAggregation}$, the detection performance remains constant across different $N_{cls}$ values. This observation aligns with the intuitive expectation that all cluster nodes, including head nodes, participate in online learning regardless of the cluster count. On the other hand, when $\beta$ takes values of either $\beta = \text{Random}$ or $\beta = \text{MaxLabels}$, detection performance becomes intricately linked to the parameter $\eta$. For instance, with $\eta = 2$, a $N_{cls} = 2$ configuration implies the participation of four nodes in online learning. In contrast, for $N_{cls} = 4$, eight nodes engage in the learning process. This correlation results

in an enhanced detection performance with an increased number of clusters.

Different Selection Strategy $(\beta)$ with Varying $(\eta)$: As illustrated in Figures 13, 14, when $\beta = Random$, increasing $\eta$ slightly enhances detection performance but consistently falls short of the detection performance achieved with $\beta = FullAggregation$. Similarly, with $\beta = MaxLabels$, increasing $\eta$ notably improves detection performance. Furthermore, we observed that with 16-25% fewer participating nodes, $\beta = MaxLabels$ outperforms $\beta = FullAggregation$. This threshold’s variability, contingent on traffic density, is evident in the transition from $\eta = 4$ to $\eta = 5$ under $\alpha = 30$, $N_{cls} = 2$, where detection performance drops, compared to the continuous increase with $\alpha = 100$, $N_{cls} = 2$.

E. Limitations of the Study

In our study, we exchanged the complete detection model (6.2MB) during online model training for both ClassicalFL and CFL approaches. However, in practical settings, object detection models may exhibit larger sizes, prompting the necessity for model compression to enhance efficiency. One limitation involves the requirement for image data captured by multiple vehicles in close proximity under similar environmental conditions. We addressed this limitation by generating synthetic datasets using the CARLA simulator [29], although real-world data would offer a more precise representation. To ensure greater traceability, we restricted the total number of participating vehicles to 12, but evaluating AR-CFL in a broader scenario would be recommended.

Notably, security or privacy-preserving mechanisms were not incorporated into this work. For a more comprehensive approach, integrating encryption and privacy-preserving techniques such as differential privacy is advisable. These considerations represent important directions for future research and refinement of the proposed framework.

VI. CONCLUSION AND FUTURE WORK

In this paper, we presented AR-CFL, an innovative framework for adaptive and Resource-aware Clustered Federated Learning. Our framework is designed specifically considering the factors impacting continuous online federated learning and communication networks in vehicular environments. Utilizing our framework, we conducted a comprehensive investigation into the scenario of online training of perception models in vehicular networks on non-IID data. To achieve this objective, we created three synthetic image datasets representing different traffic densities using the CARLA simulator. In contrast to existing literature, we addressed the constraint of limited storage on vehicles by limiting the utilized data for training to only freshly captured data at each global iteration. Our analysis revealed that increasing the traffic density enhances the detection performance of the train model. We compared the Clustering approaches against both Centralized and Classical FL learning approaches under different configurations. Furthermore, we explored the effects of varying cluster counts and different participant selection strategies within the Clustering setup. We found out that
elevating the cluster count results in an increase in cellular communication. Notably, certain participant selection strategies, such as MaxLabels, demonstrate high detection performance compared to FullAggregation approach, with up to a 25% reduction in participating nodes, and 33% less cellular communication. For future work, implementing model compression techniques could enhance the efficiency. Additionally, evaluating our framework with real-world data would be preferred to better generalize. Finally, the integration of additional encryption and privacy-preserving mechanisms such as differential privacy would offer significant benefits.

REFERENCES


