Driving Towards Efficiency: Adaptive Resource-Aware Clustered Federated Learning in Vehicular Networks

Khalil, A.; Lotfian Delouee, M.; Degeler, V.; Meuser, T.; Fernandez Anta, A.; Koldehofe, B.

DOI
10.1109/MedComNet62012.2024.10578208

Publication date
2024

Document Version
Final published version

Published in
2024 22nd Mediterranean Communication and Computer Networking Conference

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§
‖Multimedia Communications Lab, Technical University of Darmstadt, Germany
†Bernoulli Institute, University of Groningen, Netherlands
‡Informatics Institute, University of Amsterdam, Netherlands
¶IMDEA Networks Institute, Spain
§Institute of Applied Computer Science, Technische Universität Ilmenau, Germany

2024 22nd Mediterranean Communication and Computer Networking Conference (MedComNet) | 979-8-3503-9047-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/MEDCOMNET62012.2024.10578208

ABSTRACT— Guaranteeing precise perception for autonomous driving systems in diverse driving conditions requires continuous improvement and training of the perception models. In vehicular networks, *federated learning* (FL) facilitates this by enabling model training without sharing raw sensory data. Based on federated learning, *clustered federated learning* reduces communication overhead and aligns well with the dynamic nature of these networks. However, current literature on this topic does not consider critical aspects, including (1) the correlation between perception performance and the networking overhead, (2) the limited data storage on vehicles, (3) the need for training with freshly captured data, and (4) the impact of data heterogeneity (non-IID) and varying traffic densities. To fill these research gaps, we introduce AR-CFL, an *Adaptive Resource-aware Clustered Federated Learning* framework. 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 online detection model training scenario on non-IID data across varied conditions. The evaluation results highlight the robust detection performance exhibited by the trained model employing the clustered federated learning approach, despite the constraints posed by limited vehicle storage capacity. Furthermore, our study reveals that utilizing clustered federated learning enhances the training efficiency of participating nodes by up to 25% and decreases cellular communication by 33% in contrast to conventional federated learning methods.

***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 human drivers' decisions, hence increasing road safety and traffic efficiency. Reliable vehicular perception is an essential enabler of autonomous driving. Deep Neural Networks (DNNs) play a critical role in autonomous driving for vehicular perception, serving as essential models for perception tasks (e.g., object detection models). Typically, these models are centrally trained using data gathered from a limited number of vehicles operating in specific geographical areas and under certain conditions. Consequently, perception models trained on such data may exhibit low performance in object detection, even with quality management techniques in place [1], [2]. 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 [3].

Federated Learning (FL) [4] is a technique to collaboratively train models from distributed clients (e.g., road users) leveraging the computational resources of each client. In federated learning, a central server manages trained models that are updated with incremental parameter updates from participants/clients. Since the data (e.g., model updates) sent to the central server is usually much smaller than the raw training data, federated learning reduces communication requirements. Additionally, it protects (up to some level) privacy, which is important in many Internet of Things (IoT) applications (e.g., [5], [6]), by keeping the raw data stored locally.

Nevertheless, the communication demands between participants and the central server in federated learning may remain excessively high under certain conditions. Therefore, employing *clustering* can mitigate this communication overhead. In *Clustered Federated Learning*, clients are grouped into clusters based on their proximity. This approach reduces communication requirements between clients and the central server, as most of the data exchange for training occurs within clusters [7]. Additionally, it can enhance the performance of the federated learning method by reducing the time needed to train the models.

Encouraged by the advantages of clustered federated learning, this study introduces a novel Adaptive Resource-aware Clustered Federated Learning framework, named AR-CFL. This framework is specifically tailored to thoroughly investigate and optimize factors affecting online learning and communication requirements within vehicular environments. Our innovative framework integrates adaptive mechanisms to dynamically enhance the efficiency of model training. Moreover, using AR-CFL, we analyze the scenario of training an object detection model on Non-Independent and Identically Distributed (non-IID) data across diverse conditions. We systematically evaluate and

*These authors contributed equally to the paper.

discuss the results achieved through various design decisions and configuration options. In summary, this paper offers the following contributions:

1) We introduce a novel framework (AR-CFL) designed to facilitate collaborative model training tasks within vehicular environments.
2) We propose a novel concept called *Dynamic Sampling* to realistically address storage limitations of vehicles in Vehicle-to-everything (V2X) networks.
3) We present new *Dynamic Participant Selection* strategies aimed at dynamically adjusting the number of clients participating in the learning process within each cluster. Moreover, AR-CFL can change the number of clusters to meet the learning task requirements.
4) We leverage AR-CFL to investigate the scenario of online object detection model training. To achieve this, we generate three new synthetic datasets using the *CARLA* simulator, covering various conditions including different traffic densities.

The rest of the paper is organized as follows. The related work is discussed in Section II. We then explore a detailed case study of training vehicular object detection models, highlighting the need for exploiting the advantages of clustered federated learning in Section III. Following this, Section IV elaborates on the specifics of AR-CFL, explaining the system design, various algorithms, strategies, and components utilized. The evaluation scenario and discussion of results are presented in Section V. Finally, Section VI summarizes the paper and outlines future research directions.

II. Related Work

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

A. Federated Learning-based Object Detection in Vehicular Context

Extensive literature has explored the benefits of federated learning in vehicular environments, demonstrating comparable performance to traditional centralized learning methods while prioritizing user privacy [8]. Federated learning offers a range of benefits in this context, such as enhanced detection performance [9] and the ability to train models using heterogeneous data sources like vehicle onboard sensors [10]. This is achieved by adapting local training iterations dynamically and implementing model compression techniques to minimize communication overhead during model exchanges [11].

However, achieving a balance between privacy and utility involves trade-offs. For instance, opting for clients with adequate resources allows some participants to send their datasets to a central server, which can enhance utility at the cost of privacy [12]. Nonetheless, federated learning does not eliminate all privacy concerns, particularly during model exchange or client selection phases. To address this, initial solutions like multi-layer context-aware client selection and aggregation have been proposed to mitigate privacy risks [13]. These strategies help minimize privacy violations and maintain the integrity of federated learning in vehicular scenarios.

Although federated learning is proven capable of improving object detection model training [3], the amount of exchanged data and the convergence time are considerably increased to achieve an acceptable detection capability level. Hence, achieving optimum balance for both factors requires shifting from the conventional federated learning mechanisms.

B. Clustered Federated Learning in Vehicular Context

Clustered Federated Learning (CFL) presents a promising approach for addressing the complexities of federated learning in dynamic vehicular network environments. In such contexts, where client diversity varies across different network topologies, techniques like the weighted inter-cluster cycling update algorithm [14] offer adaptive strategies. Moreover, the challenges posed by imbalanced and distribution-shifted training data are being tackled through innovative clustered federated learning frameworks that group clients based on optimization direction similarities, aiming to reduce training divergence [15]. Cluster formation strategies, driven by client data distribution and considerations such as game theory principles [16] or leveraging platooning benefits [17], further highlight the potential of clustered federated learning in vehicular settings. Additionally, recent explorations into CFL-based object detection techniques in vehicular networks, leveraging Vehicle-to-Vehicle (V2V) resources to overcome communication bottlenecks [18], underscore the evolving research in this area. However, challenges persist, including the lack of adaptivity in static cluster formations and limitations in managing mobility dynamics for continuous training, emphasizing the need for further investigation and innovation in clustered FL for vehicular contexts.

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 Clustered Federated Learning (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.

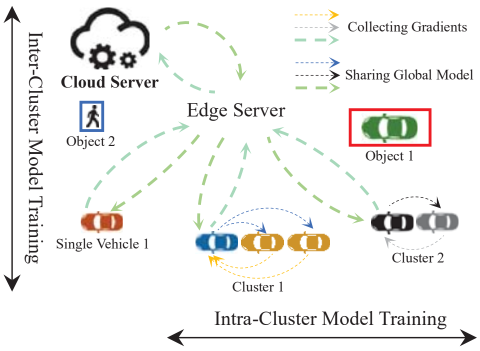


Fig. 1: An example of vehicular object detection online training by employing clustered federated learning.

- *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 illustrate the potential of leveraging a clustered federated learning approach in the context of online object detection model training to enhance vehicular perception.

As illustrated in Figure 1, the training process involves three key entities:

a) Road User: Representing vehicles or other entities that generate perception data streams about their surroundings, each road user possesses computational and communication capabilities necessary for participating in the training process. Each road user is equipped with onboard sensing configurations, such as cameras or lidar sensors [19]. For simplicity, we assume all road users deploy similar sensing technologies, generating consistent data types like camera images. Groups of road users can form clusters, with one member designated as the cluster head responsible for coordinating communication with other cluster members and the edge server for exchanging model updates.

b) Edge Server: The edge server, which can be implemented as a Roadside Unit (RSU), acts as a communication intermediary between cluster head clients and cloud servers. It functions as a basic base station, facilitating data exchange up and down the network [20], [21].

c) Cloud Server: The primary role of the cloud server is to orchestrate the entire process of object detection model training. It communicates with edge servers to distribute detection models efficiently among them.

When a cluster of vehicles is formed, a series of new learning rounds is initiated within the cluster, referred to as *Intra-Cluster Model Training*. During this phase, the trained model is exchanged directly among cluster members using V2V communication, thereby reducing reliance on long-range cellular data exchange. Additionally, stable and direct V2V communication within the cluster has the potential to shorten convergence time.

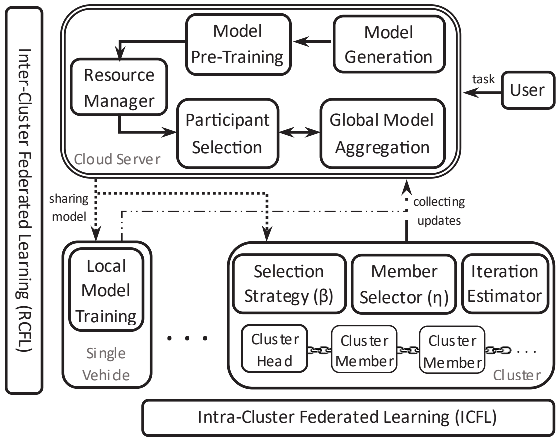


Fig. 2: The AR-CFL System Design includes the main components of both levels of federated learning.

Intra-cluster model training involves multiple training rounds within the cluster before the model is shared with the edge server, which subsequently forwards it to the cloud server for model aggregation. Following the aggregation of model updates received from edge servers, the resulting model is distributed back to the edge servers and subsequently to the clusters over several training rounds, known as *Inter-Cluster Model Training*.

Based on these elements and the foundational structure of clustered federated learning, in the next section, we introduce the AR-CFL framework.

IV. AR-CFL System Design

Figure 2 illustrates the core components of AR-CFL. AR-CFL incorporates two distinct levels of federated learning: Inter-Cluster Federated Learning (RCFL) and Intra-Cluster Federated Learning (ICFL).

In RCFL, the interaction flow occurs between the cloud server (located in the cloud layer), edge servers, and the participants (i.e., vehicles). When a user (e.g., car manufacturer) starts a model training task, the cloud server initiates a global model for online training. This global model is pre-trained using a random sample of vehicle data. Subsequently, the cloud server determines the requisite number of vehicles needed for the specific task, considering the required amount of data and computing resources for execution. The Resource Manager module adjusts control variables β (in-cluster participant selection strategy) and η (number of in-cluster participants) to align with the necessary Computing Resources (CR) for each task. Together, these parameters dictate the total number of participants N_n and thus, the number of clusters. A comprehensive list of setup and evaluation parameters utilized in our study can be found in Table I.

On the other hand, ICFL involves the learning process within each cluster. Here, the cluster head determines the necessary number of local iterations (i.e., training rounds within the cluster) based on the cluster's available computing resources.

In the following, we provide detailed information about both federated learning levels:

A. Inter-Cluster Federated Learning (RCFL)

Algorithm 1 outlines the online learning process from the perspective of the cloud server. Here, we omit the edge server from our analysis as it only facilitates the data exchange up and down the network. Initially, the cloud server generates the shared global model ω_g, where g denotes the global iteration number. At the beginning of the RCFL process, the cloud server initializes the list of available clients (i.e., vehicles) and defines the minimum required resources for the task. Furthermore, the cloud server pre-trains the global model ω_0 using an initial small dataset. Additionally, the cloud server determines the participant selection strategy β within each cluster and communicates this strategy to the cluster heads to ensure uniform selection across all clusters. Various selection strategies are introduced and detailed in Section IV-B. Moreover, the number of participants chosen according to the specified strategy is determined using the value η. The cloud server selects clients with the highest Computing Resources (CR) values (i.e., $CR_{c_{max}}$) until their cumulative computing resources (R_g) meet the required threshold for the task. Notably, fairness among selected clients can be achieved by integrating specific scheduling algorithms [22], [23]. However, such integration is omitted in our framework for the sake of simplicity. Each selected participant receives the global model ω_g and updates its current local model ω_g^c. If the selected client is a cluster of vehicles, the second level of federated learning is triggered within the cluster (refer to Algorithm 2). Ultimately, the cluster head uploads the aggregated local model of the cluster to the cloud server. Conversely, if the client is an individual vehicle, it conducts E_c local training iterations on the local model. This process generates the updated local model ω_{g+1}^c, which is then uploaded to the cloud server. Finally, the cloud server aggregates the received trained local models using a weighted sum to produce the new global shared model ω_{g+1}.

B. Intra-Cluster Federated Learning (ICFL)

The second level of federated learning is detailed in Algorithm 2. Following the initialization of the cluster member set and the local training model, during each cluster training iteration k, the cluster head selects participants based on the in-cluster participant selection strategy (denoted as β) and the parameter η. There are three distinct selection strategies for choosing in-cluster participants represented by β: Full Aggregation, Random, and MaxLabels. In the Full Aggregation strategy ($\beta = FullAggregation$), all cluster members participate, and their models are aggregated at the cluster head. Conversely, in the Random strategy ($\beta = Random$), participants are selected randomly during each iteration. In the MaxLabels strategy ($\beta = MaxLabels$), cluster members with the highest number of labels (data-rich) are chosen (refer to Figure 3).

In both the Random and MaxLabel strategies, η determines the number of members involved in local training within each cluster. The cloud server determines η based on various conditions, such as the task at hand and the

Algorithm 1 RCFL Procedure

Initialization:
    $\mathcal{C} \leftarrow \{[c_1, CR_{c_1}], ..., [c_n, CR_{c_n}]\}$; ▷ clients
    $CR_{task} \leftarrow$ Minimum Required Resources for *task*;
    Pre-trained ω_0; ▷ Initial model
    $\beta \leftarrow$ In-Cluster Participant Selection Strategy;
    $\eta \leftarrow$ Number of In-Cluster Participants;
for global iteration g = 0, 1, ... **do**
    $Update(\mathcal{C})$;
    $R_g \leftarrow 0$; ▷ Sum of Computing Resources
    $C_g \leftarrow \varnothing$; ▷ Selected Clients
    $\mathcal{C}' \leftarrow$ Sort $(\mathcal{C}, CR)$; ▷ Desc Sorted Client Set by CR
    while $R_g \leq CR_{task}$ **do**
        $c_{max} \leftarrow \mathcal{C}'[0]$; ▷ Client with the highest CR value
        $C_g \leftarrow C_g \cup c_{max}$;
        $R_g \leftarrow R_g + CR_{c_{max}}$
        $\mathcal{C}' \leftarrow \mathcal{C}' - \mathcal{C}'[0]$;
    Distribute ω_g to clients in C_g; ▷ Global model
    for client $c \in C_g$ **do** ▷ In Parallel
        $\omega_g^c \leftarrow \omega_g$;
        if c is a cluster **then** ▷ In-cluster training
            E_c , $\omega_{g+1}^c \leftarrow$ ICFL(ω_g^c, β, η); ▷ Algorithm 2
        else ▷ Single client training
            $E_c \leftarrow E_l$
            $\omega_{g+1}^c \leftarrow$ LocalTraining (ω_g^c, E_c);
    $E_\sigma \leftarrow \sum_{c \in C_g} E_c$;
    $\omega_{g+1} \leftarrow \sum_{c \in C_g} (E_c/E_\sigma) \times \omega_{g+1}^c$; ▷ Trained model

(a) Image contains one label. (b) Image contains four labels.

Fig. 3: Example of two image samples. Here, the image in (b) is more data-rich than the image in (a).

computation capabilities of the vehicles. For instance, setting $\eta = 2$ implies that two clients from each cluster will participate in the local iteration. When $\eta = 2$ and $\beta = MaxLabels$, the two cluster members with the largest label counts in that specific cluster are selected.

In the case of the FullAggregation setup, η equals the total number of cluster members, including the cluster head. Subsequently, each selected member trains the model with freshly captured data and submits its update to the cluster head. The collected updates are then aggregated into a new model and used for the subsequent cluster iteration. In our configuration, we specify the number of cluster iterations as $k = 1$. Upon completion of the training rounds, the cluster head forwards the last aggregated update to the edge server and subsequently to the cloud server as a result of this round of the global training procedure (RCFL).

C. Handling The Limited Storage on Vehicles

In contrast to conventional federated learning approaches that rely on large, static data splits across clients, our method handles the constrained storage capacity of participating vehicles by systematically freeing up utilized data at the end of each training round. This practice ensures space for acquiring fresh data, thereby enhancing

Algorithm 2 ICFL Procedure

Initialization:
  Input: ω_g^c, β, η
  $Cluster \leftarrow \{[m_1, CR_{m_1}], ..., [m_n, CR_{m_n}]\}$;
  $\omega_0^{Cluster} \leftarrow \omega_g^c$; ▷ Initial model in cluster
  $E_c \leftarrow \varnothing$
for cluster iteration l = 0, 1, ..., k **do**
  $C_l \leftarrow \varnothing$; ▷ Selected members for training
  if $\beta == Full\_Aggregation$ **then**
    $C_l \leftarrow Cluster$;
  else if $\beta == Random$ **then**
    for $i \leftarrow 1$ to η **do**
      $c_i \leftarrow Random(Cluster)$;
      $C_l \leftarrow C_l \cup c_i$;
      $Cluster \leftarrow Cluster \setminus \{c_i\}$;
  else if $\beta == MaxLabel$ **then**
    for $i \leftarrow 1$ to η **do**
      $c_i \leftarrow \max_{m_j \in Cluster} LabelCount_{m_j}$;
      $C_l \leftarrow C_l \cup c_i$;
      $Cluster \leftarrow Cluster \setminus \{c_i\}$;
  Distribute $\omega_l^{Cluster}$ to clients in C_l;
  for $c' \in C_l$ **do** ▷ In Parallel
    $\omega_l^{c'} \leftarrow \omega_l^{Cluster}$;
    $\omega_{l+1}^{c'} \leftarrow$ LocalTraining ($\omega_l^{c'}$, E_l);
    $\Omega \leftarrow \Omega \cup \omega_{l+1}^{c'}$; ▷ Set of Collected Updates
    E_c = E_c + E_l
  $\omega_{l+1}^{Cluster} \leftarrow Aggregate(\Omega)$;
return E_c, $\omega_k^{Cluster}$; ▷ Total number of local training iterations and final in-cluster trained model

the realism and effectiveness of our approach, especially within the dynamic and resource-constrained environment of vehicular federated learning scenarios. This process ensures that the data utilized by participating vehicles to train local models in one global iteration is not retained for use in subsequent global iterations. Our method optimizes the efficiency of model training by accounting for the inherent limitations of onboard storage and considering the dynamically changed conditions surrounding participating vehicles.

V. Evaluation

In this section, we explore the scenario of online object detection model training using our AR-CFL framework. 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 different approaches.
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).

Our investigation focuses on two key aspects: *online learning efficiency* and *communication overhead*. The evaluation covers the following:

TABLE I: Evaluation parameters with descriptions.

Parameter	Description
β	In-cluster member selection strategy
η	Number of participants in cluster
N_v	Total number of vehicles
N_n	Number participating vehicles
α	Traffic density (30, 50, 100)
$Ncls$	Number of clusters
E_g	Number of global iterations (epochs)
g	Global iterations index
E_l	Number of local training iterations
B	Batch size
$lr0, lrf$	Learning rate parameters
tr_t	Total training time
e_d	Total volume of exchanged data for training
ed_l	Data volume (cellular communication)
ed_s	Data volume (direct V2V communication)

1) We analyze how varying traffic density influences the overall system performance.
2) We investigate how clustered federated learning enhances the efficiency of online learning in terms of communication overhead and learning efficiency, as compared to the centralized learning approach and the classical federated learning approach.
3) We explore the influence of various selection strategies, the number of participants in the cluster, and the number of clusters on the overall performance.

A. Evaluation Scenario and Experimental Setup

We outline now the evaluation scenario and experimental setup aimed at assessing the different approaches. Our focus lies in training an object 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 [24]. The experiments employed the *YoloV8n* model [25] as a 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 day-time, rainy weather day-time, clear weather night-time, and rainy weather night-time. Experiments were conducted using a balanced combination of these conditions.

2) General Setup Variables: The total number of vehicles eligible to participate in the training process is fixed at $N_v = 12$. We consider different traffic densities denoted by α, specifically with values of 30, 50, and 100, where $\alpha = 50$ signifies the presence of 50 vehicles in the environment. Notably, these vehicles are separate from the $N_v = 12$ vehicles eligible to participate in the model training process. Furthermore, we investigate various scenarios by adjusting the number of clusters (N_{cls}) to either 2 or 4, evenly distributing the clients across the clusters.

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.

(a) clear day-time (b) rainy day-time

(c) clear night-time (d) rainy night-time

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.

- *ClassicalFL*: In this case, no clustering is considered, and all federated learning 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 $B = 16$. We established the learning rate parameters with $lr0$ and lrf, configured at their default values of $lr0 = lrf = 0.01$. Additionally, we selected $optimizer = auto$ while maintaining default values for all other model training and validation parameters [26].

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 used to evaluate the detection model's performance are:

- mean Average Precision (mAP): This metric considers precision and recall across multiple object classes [27]. 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_t 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 clustered federated learning 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:

$$e_d = \sum_{i=1}^{N_v} \sum_{j=1}^{k_i} data_s(i, j)$$

where k_i is the number of data chunks collected in vehicle *i*, and $data_s(i, j)$ 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_n 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:

$$e_d = \sum_{g=1}^{E_g} 2 \cdot N_n \cdot model_s$$

where $model_s$ indicates the model size.

Finally, we consider two-level aggregation while computing data exchange volumes in the different clustered federated learning 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

$$ed_l = \sum_{g=1}^{E_g} 2 \cdot N_{cls} \cdot model_s$$

replacing N_n with N_{cls}.

On the other hand, direct V2V communication is required between cluster heads and members is faster and less costly. The bandwidth cost for this communication type, denoted as ed_s, is calculated as

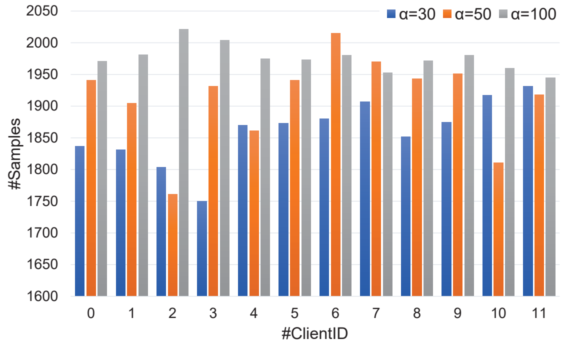
$$ed_s = \sum_{g=1}^{E_g} 2 \cdot N_{cls} \cdot \eta \cdot model_s$$

Finally, the value of e_d is computed by summing the ed_l and ed_s values as follows:

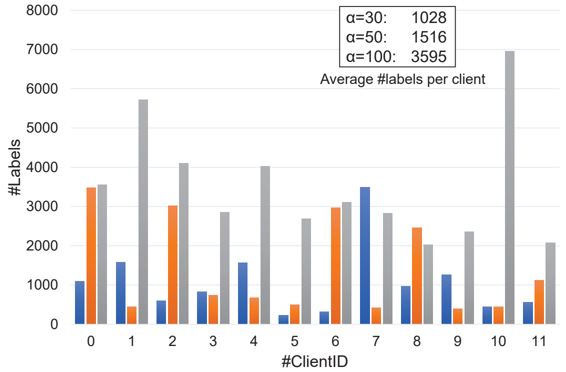
$$e_d = ed_l + ed_s$$

C. Data Generation

We used the CARLA simulator [24] to generate training and validation data, building upon [28] for concurrent image and ground truth data generation from multiple vehicles. We split the data into 80% for training and 20% for validation. Simulations used the pre-built Map-Town04 [29], and data for each detection task was uniformly generated.



(a) Sample distribution over clients (vehicles).



(b) Label distribution over clients (vehicles).

Fig. 5: Data distribution statistics of the clients $N_v = 12$. We generated three datasets with varying traffic densities $\alpha = 30, 50, 100$.

1) Data Distribution Statistics: Figure 5a visually illustrates sample (image) distribution across clients for different traffic densities α, totaling 22,327 for $\alpha = 30$, 22,948 for $\alpha = 50$, and 23,715 for $\alpha = 100$. Differences in sample sizes are relatively small compared to the complete dataset.

Figure 5b depicts label (car bounding box) distribution across the clients' samples for different traffic densities across all iterations. Higher α values result in increased total label count.

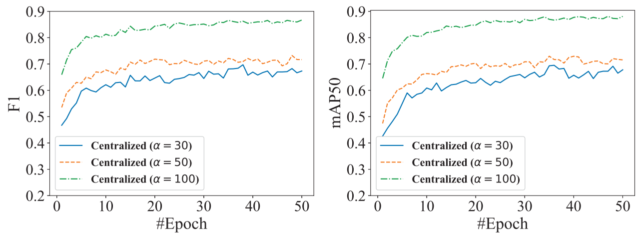
An important observation from Figure 5 is the minimal variation in data quantity among clients, suggesting that the difference in sample size can be considered insignificant. However, a noticeable label distribution imbalance across clients highlights our consideration of non-IID data handling and illustrates data heterogeneity across clients [30], [31].

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.

TABLE II: Total training time tr_t of different approaches

Approach	**Training time** ($\mathbf{tr}_t$)
Centralized	207 minutes
ClassicalFL	101 minutes
Clustering	101 minutes



(a) F1 Score (b) Mean Average Precision

Fig. 6: The detection performance of the *Centralized* approach is evaluated across different traffic densities with values $\alpha = 30, 50, 100$. #Epoch refers to the number of global iterations. A noticeable enhancement in performance is evident with the increase in traffic density.

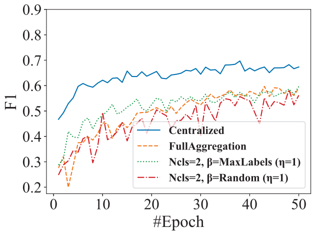
Notably, increased traffic density correlates with increased detection performance without influencing training time or data exchange volume. These aspects depend solely on sample size and not on the characteristics within the samples. These observations go beyond the *Centralized* approach and are applicable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

2) Influence of Clustering vs. Centralized Approach on Online Learning Efficiency: Examining the impact of clustered federated learning 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 clustered federated learning setups in detection performance. Despite this, the gap remains constant across different α values.

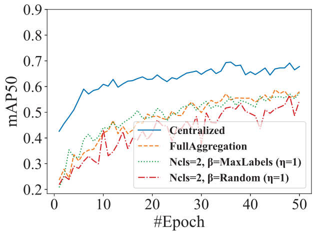
The true advantage of both federated learning and clustered federated learning emerges in reduced training time. As illustrated in Table II, both approaches demonstrate an impressive 52% decrease in training time compared to *Centralized* training approach. The reason behind that is the increased computational resources made accessible through federated learning approaches, allowing multiple clients to concurrently engage in training using their local data partitions. It is important to note that when assessing training times, we assume comparable computational power between the central server and individual vehicles. Furthermore, as illustrated in Figure 8, the clustered federated learning 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 clustered federated learning 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* participants selection strategies of clustered federated learning outperform traditional *ClassicalFL* in detection performance. *FullAggregation* involves all clients in each

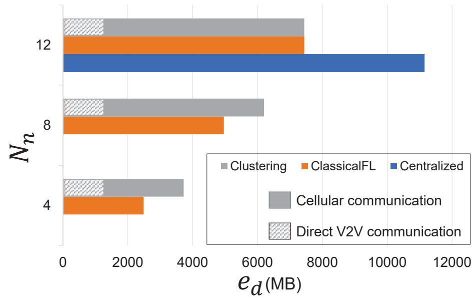


(a) F1 Score

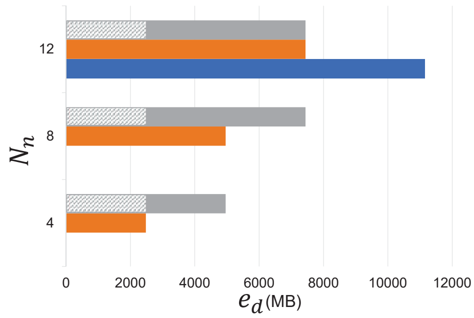


(b) Mean Average Precision

Fig. 7: Comparing the detection performance between the *Centralized* approach and selected *Clustering* approaches under a traffic density of $\alpha = 30$.



(a) $N_{cls} = 2$

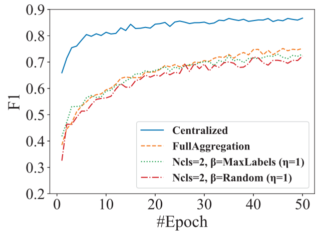


(b) $N_{cls} = 4$

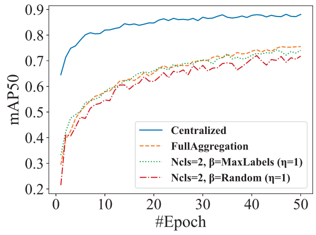
Fig. 8: Comparison of exchanged data volume across different approaches relative to the number of participating clients N_n and the number of clusters N_{cls}. All clustered federated learning setups exhibit consistent values across varying traffic densities.

iteration, contrasting with *ClassicalFL*, which randomly selects a subset (N_n) of clients per iteration. However, *FullAggregation* introduces increased short-range communication (ed_s) 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, clustered federated learning introduces additional communication overhead. Nevertheless, with the increased number of participants, there is a decrease in ed_l and an increase in ed_s, highlighting the benefits of clustered federated learning over *ClassicalFL*. The $Random$ setup of clustered federated learning with $\eta = 1$ shows comparable performance to *ClassicalFL*, randomly selecting clients in each iteration. With increased η (e.g., $\eta = 3$), $Random$ strategy in

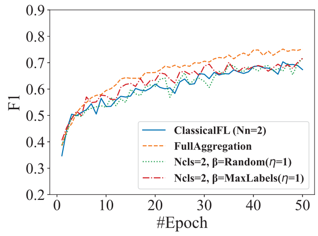


(a) F1 Score

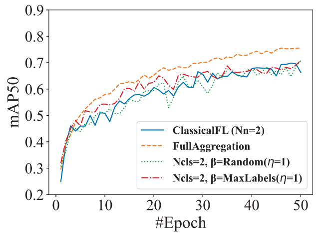


(b) Mean Average Precision

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



(a) F1 Score



(b) Mean Average Precision

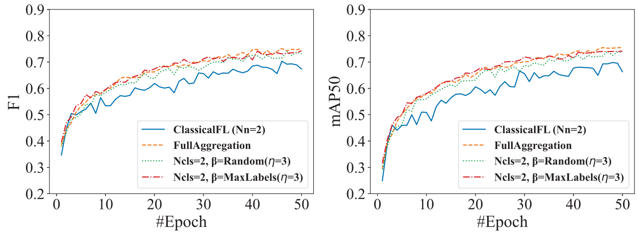
Fig. 10: Comparing the detection performance between the *ClassicalFL* approach and selected *Clustering* approaches under a traffic density of $\alpha = 100$ with one selected client at each cluster ($\eta = 1$).

clustered federated learning outperforms *ClassicalFL* in detection performance due to more participating clients.

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

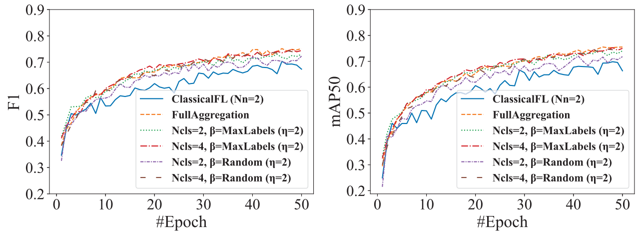
Varying Cluster Numbers (N_{cls}): As shown in Table II, the training time consistently stands at $tr_t = 101$ minutes across varying numbers of clusters N_{cls}. Examining the communication overhead illustrated in Figure 8, we observe an increase in long-range communication overhead corresponding to 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 $\beta = FullAggregation$, 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. It is noteworthy that in the context of the *ClassicalFL* approach, particularly when $N_n = 12$, it yields perception capability results identical to those in the *Clustering* setup with $\beta = FullAggregation$. On the



(a) F1 Score (b) Mean Average Precision

Fig. 11: Comparing the detection performance between the *ClassicalFL* approach and selected *Clustering* approaches under a traffic density of $\alpha = 100$ with three selected client at each cluster ($\eta = 3$).



(a) F1 Score (b) Mean Average Precision

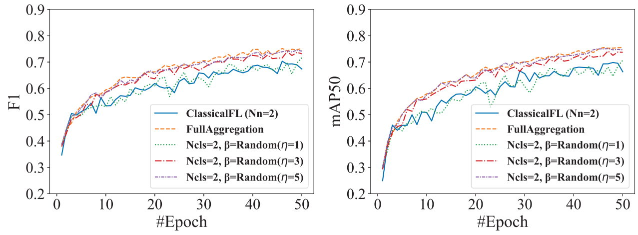
Fig. 12: The detection performance of various *Clustering* approaches under a traffic density of $\alpha = 100$, with two selected clients in each cluster ($\eta = 2$), and varying cluster counts N_{cls}.

other hand, when β takes values of either $\beta = Random$ or $\beta = MaxLabels$, detection performance becomes intricately linked to the parameter η. 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 (β) with Varying (η): As illustrated in Figures 13, 14, when $\beta = Random$, increasing η slightly enhances detection performance but consistently falls short of the detection performance achieved with $\beta = FullAggregation$. Similarly, with $\beta = MaxLabels$, increasing η 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 $\alpha = 100$, $N_{cls} = 2$.

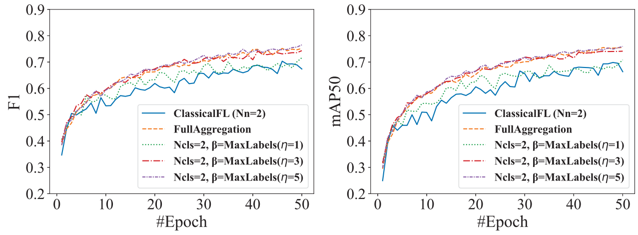
E. Limitations of the Study

In our study, we exchanged the complete detection model (6.2MB) during online model training for both *ClassicalFL* and *clustered federated learning* 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 vehi-



(a) F1 Score (b) Mean Average Precision

Fig. 13: Comparing the detection performance between the *FullAggregation* and *Random* approaches of clustered federated learning under a traffic density of $\alpha = 100$, with a varying number of selected clients at each cluster ($\eta = 1, 3, 5$), and two clusters ($N_{cls} = 2$).



(a) F1 Score (b) Mean Average Precision

Fig. 14: Comparing the detection performance between the *FullAggregation* and *MaxLabels* approaches of cluster federated learning under a traffic density of $\alpha = 100$, with a varying number of selected clients at each cluster ($\eta = 1, .., 5$), and two clusters ($N_{cls} = 2$).

cles in close proximity under similar environmental conditions. We addressed this limitation by generating synthetic datasets using the CARLA simulator [24], 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 object detection model online training 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 training the local models using 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 Federated Learning* approaches against both *Centralized* and *Classical Federated Learning* approaches under different configurations. Furthermore, we explored the effects of varying cluster counts and different participant selection strategies within the clustered federated learning setups. We found out that increasing the number of clusters increases long-range 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 long-range 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 generalization. Finally, the integration of additional encryption and privacy-preserving mechanisms such as differential privacy can offer significant benefits.

VII. Acknowledgement

This work has been funded by the German Research Foundation (DFG) within the Collaborative Research Center (CRC) 1053 MAKI, as part of subprojects B1 and C2.

References

[1] M. Lotfian Delouee, B. Koldehofe, and V. Degeler, "Aqua-cep: Adaptive quality-aware complex event processing in the internet of things," in *Proceedings of the 17th ACM International Conference on Distributed and Event-Based Systems*, 2023, p. 13–24.

[2] ——, "Towards adaptive quality-aware complex event processing in the internet of things," in *2022 18th International Conference on Mobility, Sensing and Networking (MSN)*, 2022, pp. 571–575.

[3] A. Khalil, T. Meuser, Y. Alkhalili, A. Fernández Anta, L. Staecker, R. Steinmetz *et al.*, "Situational collective perception: Adaptive and efficient collective perception in future vehicular systems," in *International Conference on Vehicle Technology and Intelligent Transport Systems*. SCITEPRESS Science and Technology Publications, 2022, pp. 346–352.

[4] C. Zhang and et al., "A survey on federated learning," *Knowledge-Based Systems*, vol. 216, p. 106775, 2021.

[5] M. Lotfian Delouee, B. Koldehofe, and V. Degeler, "Poster: Towards pattern-level privacy protection in distributed complex event processing," in *Proceedings of the 17th ACM International Conference on Distributed and Event-Based Systems*, 2023, p. 185–186.

[6] M. Lotfian Delouee, V. Degeler, P. Amthor, and B. Koldehofe, "App-cep: Adaptive pattern-level privacy protection in complex event processing systems," in *10th International Conference on Information Systems Security and Privacy (ICISSP'24)*. SCITEPRESS – Science and Technology Publications, Dec. 2023.

[7] Z. He, L. Wang, and Z. Cai, "Clustered federated learning with adaptive local differential privacy on heterogeneous iot data," *IEEE Internet of Things Journal*, vol. 11, no. 1, pp. 137–146, 2024.

[8] D. Jallepalli, N. C. Ravikumar, P. V. Badarinath, S. Uchil, and M. A. Suresh, "Federated learning for object detection in autonomous vehicles," in *2021 IEEE Seventh International Conference on Big Data Computing Service and Applications (BigDataService)*, 2021, pp. 107–114.

[9] G. Rjoub, O. A. Wahab, J. Bentahar, and A. S. Bataineh, "Improving autonomous vehicles safety in snow weather using federated yolo cnn learning," in *Mobile Web and Intelligent Information Systems*, J. Bentahar, I. Awan, M. Younas, and T.-M. Grønli, Eds. Cham: Springer International Publishing, 2021, pp. 121–134.

[10] T. Zheng, A. Li, Z. Chen, H. Wang, and J. Luo, *AutoFed: Heterogeneity-Aware Federated Multimodal Learning for Robust Autonomous Driving*. Association for Computing Machinery, 2023.

[11] Y. Chen, C. Wang, and B. Kim, "Federated learning with infrastructure resource limitations in vehicular object detection," in *2021 IEEE/ACM Symposium on Edge Computing (SEC)*, 2021, pp. 366–370.

[12] A. M. Elbir, S. Coleri, A. K. Papazafeiropoulos, P. Kourtessis, and S. Chatzinotas, "A hybrid architecture for federated and centralized learning," *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 3, pp. 1529–1542, 2022.

[13] X. Zhou, W. Liang, J. She, Z. Yan, I. Kevin, and K. Wang, "Two-layer federated learning with heterogeneous model aggregation for 6g supported internet of vehicles," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 5308–5317, 2021.

[14] B. Li, Y. Jiang, W. Sun, W. Niu, and P. Wang, "Fedvanet: Efficient federated learning with non-iid data for vehicular ad hoc networks," in *2021 IEEE Global Communications Conference (GLOBECOM)*, 2021, pp. 1–6.

[15] M. Duan, D. Liu, X. Ji, Y. Wu, L. Liang, X. Chen, Y. Tan, and A. Ren, "Flexible clustered federated learning for client-level data distribution shift," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 11, pp. 2661–2674, 2022.

[16] R. Lu, W. Zhang, Y. Wang, Q. Li, X. Zhong, H. Yang, and D. Wang, "Auction-based cluster federated learning in mobile edge computing systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 4, pp. 1145–1158, 2023.

[17] C. Quadri, V. Mancuso, V. Cislaghi, M. A. Marsan, and G. P. Rossi, "From plato to platoons," in *2021 19th Mediterranean Communication and Computer Networking Conference (MedComNet)*, 2021, pp. 1–8.

[18] A. Taïk, Z. Mlika, and S. Cherkaoui, "Clustered vehicular federated learning: Process and optimization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 25 371–25 383, 2022.

[19] L. Crosara, M. Brocco, C. Cavalagli, X. Wu, E. Gindullina, and L. Badia, "Data injection in a vehicular network framed within a game theoretic analysis," in *2023 21st Mediterranean Communication and Computer Networking Conference (MedComNet)*, 2023, pp. 25–28.

[20] F. Busacca, C. Cirino, G. Faraci, C. Grasso, S. Palazzo, and G. Schembra, "Multi-layer offloading at the edge for vehicular networks," in *2020 Mediterranean Communication and Computer Networking Conference (MedComNet)*, 2020, pp. 1–8.

[21] D. et al., "Roadmap for edge ai: a dagstuhl perspective," *SIGCOMM Comput. Commun. Rev.*, vol. 52, no. 1, p. 28–33, mar 2022.

[22] J. Zhang and et al., "Joint scheduling of participants, local iterations, and radio resources for fair federated learning over mobile edge networks," *IEEE Transactions on Mobile Computing*, vol. 22, no. 7, pp. 3985–3999, 2023.

[23] A. Sultana and et al., "Eiffel: Efficient and fair scheduling in adaptive federated learning," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 12, pp. 4282–4294, 2022.

[24] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *In CoRL'17*, 2017, pp. 1–16.

[25] G. Jocher, A. Chaurasia, and J. Qiu, "YOLO by Ultralytics," Jan. 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[26] "Ultralytics documentation: Train mode arguments," 2023. [Online]. Available: https://docs.ultralytics.com/modes/train/#arguments

[27] R. Padilla, S. L. Netto, and E. A. B. da Silva, "A survey on performance metrics for object-detection algorithms," in *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, 2020, pp. 237–242.

[28] AveesLab, "Carfree: Automatic ground truth generation in carla," 2021. [Online]. Available: https://github.com/AveesLab/CarFree#automatic-ground-truth-generation-in-carla

[29] CARLA Simulator, "Carla documentation - map town04," 2022. [Online]. Available: https://carla.readthedocs.io/en/latest/map_town04/

[30] P. Kairouz and et al, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021. [Online]. Available: http://dx.doi.org/10.1561/2200000083

[31] A. Khalil, A. Wainakh, E. Zimmer, J. Parra-Arnau, A. F. Anta, T. Meuser, and R. Steinmetz, "Label-aware aggregation for improved federated learning," in *2023 Eighth International Conference on Fog and Mobile Edge Computing (FMEC)*, 2023, pp. 216–223.