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

ABSTRACT— Achieving precise perception models for fully autonomous driving in diverse driving conditions requires continuous online model training. In vehicular networks, *federated learning* (FL) facilitates this by enabling model training without sharing raw sensory data. Additionally, *clustered FL* reduces communication overhead and aligns well with the dynamic nature of these networks. However, current literature on this topic overlooks critical dimensions, including (1) the correlation between application efficiency 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 contributions 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, and addresses the scenario of online car detection model training on non-IID data under diverse conditions. Empirical evaluations of AR-CFL reveal a high perception performance of the optimized model, measured by metrics such as F1 score and mean average precision, compared to some classical FL scenarios, even with up to a 25% less participating nodes, and a 33% reduction in long-range communication.

Index Terms—**Vehicular Networks, Federated Learning, Adaptivity, Clustering, Resource Efficiency, Collective Perception, Deep Neural Networks.**

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 throughput [1]. However, today's vehicles have a narrow perception of the environment due to limited onboard sensing [2]. To cope with this, exchanging collected data among vehicles (and all road users) can help achieve a better perception of the environment [3]. The evolving Vehicle-to-everything (V2X) technologies provide means of communication between road users and enable them to collect and aggregate perception data cooperatively, in the so-called Collective Perception (CP) [4].

Deep Neural Networks (DNNs) have a pivotal role for individual perception and object detection in autonomous driving. Currently, these networks usually undergo centralized training before deployment, utilizing data that is limited in its coverage of various situations. Consequently, DNN models trained on such data may exhibit low performance in object detection, even with quality management techniques in place [5], [6]. 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) is a technique to train DNN models from distributed data sources (e.g., using the computational resources of each road user) [7]. 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 IoT applications (e.g., [8]), by keeping the raw data stored locally.

Recent studies demonstrate that FL techniques can train Deep Neural Networks (DNNs) with optimized client-server interactions [9]. Moreover, FL is a good paradigm for implementing cooperative perception techniques among heterogeneous resources with non-IID data (i.e., when the data from different parties has different characteristics) [10], [11]. With these benefits in mind, continually training a DNN employing FL (e.g., [12]) 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 in close proximity into a cluster,

*Co-first authors.

and having a cluster head collecting the updates from the cluster members, aggregating them, and sending the resulting update to the central server [13]. Clustered FL requires extensive coordination and synchronization between the involved entities, but it reduces the communication requirements between clients and the central server, since most of the training data exchange occurs inside clusters [14]. Moreover, it can help enhance the performance of the FL approach by decreasing the time required to train the model up to an acceptable 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 conduct a thorough investigation of training 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, downsizing 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 supports the dynamic adjustment of the client set participating in the learning process within 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, which 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. We provide an overview of the AR-CFL system model in Section IV. Then, Section V presents the details of AR-CFL. The evaluation results of AR-CFL are exhibited in Section VI. Finally, Section VII 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, followed by pointing the research gaps.

A. FL-based Object Detection in Vehicular Context

The effects of employing FL-based techniques in vehicular environments have been explored vastly in the literature. For example, utilizing FL can yield performance comparable to traditional centralized deep learning while preserving user's privacy locally [15]. Besides, the detected objects should possess an acceptable level of perception capability, which is essential to the success of vehicular network applications (e.g., autonomous driving [16]). Moreover, heterogeneity in the data sources (i.e., vehicle's onboard sensors) makes the aggregation function in the FL's central server even more complex at the end of each iteration [17]. Although employing FL reduces the required communicated data in the learning process compared to the centralized predecessors, resource allocation and sensor deployment still remain challenging. However, it is proved that by providing network management with a multi-layer graph, such challenges can be overcome [18]. Also, as a solution for varying client resources, dynamically adjusting local training iterations and using model compression to reduce communication overhead during model exchanges have been introduced in the literature [19].

Even though one of the critical motivations for employing FL is to preserve privacy, some violations are still possible in the model exchange phase and client selection. However, a multi-layer context-aware client selection and aggregation can be utilized that degrades privacy violations through encryption [20]. On the other hand, establishing trade-offs has been studied to trade some privacy for achieving higher utility, e.g., a hybrid FL model was introduced by opting for clients with sufficient resources, while others send their datasets to the central server [21].

B. Clustered Federated Learning in Vehicular Context

The idea of forming clusters of clients in an FL-based learning technique in vehicular networks has been briefly studied previously. In this vein, a novel approach for cluster formation based on client data distribution has been introduced by incorporating game theory principles for client selection within clusters [22]. To adapt to the dynamic client diversity in different VANET topologies, a hierarchical inner-cluster Federated Learning model has been presented alongside a weighted inter-cluster cycling update algorithm [23]. 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 [24]. Although CFL-based object detection techniques in vehicular networks have been studied briefly (cf., [25]), static formation of clusters of vehicles (e.g., only those that are under the coverage of a base station) and the limitation of dealing with mobility of vehicles (i.e., vehicles handover) to enable continuous training indicate further studies are required.

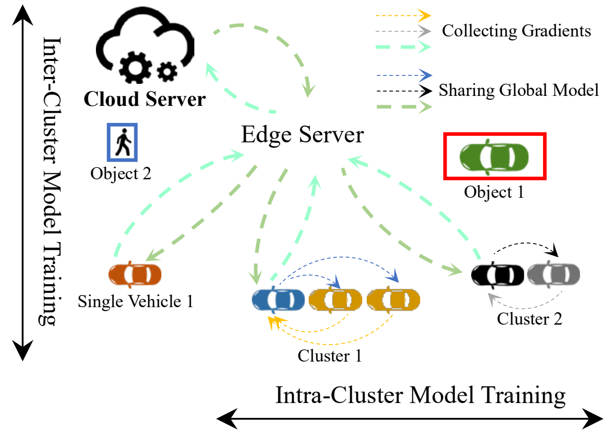


Fig. 1: An example of improving object detection's performance by employing a two-level federated learning.

C. Research Gaps

Upon reviewing existing studies and frameworks, it becomes apparent that several significant research gaps warrant further investigation. The specific areas for potential research directions are delineated as follows:

Application vs. 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 storage on vehicles, a non-realistic assumption that leads to an artificial 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 encompassed the exploration of varying cluster counts and the involvement of a diverse number of vehicles within clusters.

In this work, we endeavor to provide a solution, so-called AR-CFL, that simultaneously considers the mentioned research gaps.

III. CASE STUDY: ONLINE OBJECT DETECTION MODEL TRAINING

In Figure 1, we illustrate an online training of object detection scenario that aims to utilize data collected by road users in the detection procedure. The derived insights can be helpful in real-world scenarios such as autonomous driving. Here, we demonstrate how a system can benefit from involving the clustering concept in designing a two-level FL approach. In a conventional situational collective perception approach, it has been proved that federated learning manages to assist object detection [3]. However, one can observe that the amount of exchanged data has to be considerably increased to achieve an acceptable detection capability level. Furthermore, achieving the optimum convergence time must be considered a significant performance factor in such approaches. Hence, a modification in the conventional FL mechanism is necessary to solve the mentioned challenges.

As mentioned earlier, clustering can be employed to achieve the required perception capability sooner with less data exchange. In more detail, when a cluster is selected as an FL client, a new set of learning rounds will be initiated within the cluster. This inside-cluster learning process is called Intra-Cluster Federated Learning. The updated model (i.e., the outcome of inside-cluster training) improves the trained model's perception capability in the upper level of FL. First, a multiple-round trained model is collected from cluster members, which enhances the perception level of the aggregated global model. Second, less data will be exchanged between FL participants and the central server. Instead, intra-cluster communicated data will be added to the total exchanged data, which is negligible. Moreover, the convergence time is decreased due to stable and short communication within the cluster.

IV. PRELIMINARIES

In this section, we present the system model by introducing AR-CFL components, the model of data processing, and the provided clustering model.

A. AR-CFL Model

The proposed approach comprises the main components used in collective perception scenarios (see Figure 1).

1) 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*. In our model, parked cars (i.e., illustrated in black) can also create a cluster and participate in the learning process only with other in-range parked cars. Each road user has its own onboard sensing configurations (e.g., a *camera* or *lidar* sensor embedded in a vehicle). The produced data stream is fed into the next processing stage (e.g., a machine-learning approach) for object detection and classification. By merging the list of detected objects with the spatial information, a road user can build its own *local environment model* that will be used in subsequent decision-making processes. Since we employed CP in our mechanism, road users are empowered by the ability to exchange perception data, extending spatial awareness above their own limited perception. Such perception data is encapsulated in collective perception messages [26] to be transferred between road users. Such message exchange can be performed directly between road users or through any intermediate node (e.g., an edge server) [27]. For the sake of simplicity, we assume that all road users have enough computing resources and similar sensing deployment to generate the same data type and operate with identical environment models.

2) Edge Server: In a conventional scenario, an *edge server* acts as a simple base station that only forwards data. However, we assume each edge server has the required resources for computation and communication to process data and validate the environment models [28]. Besides communicating with road users, edge servers can also exchange data with each other. 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.

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

B. Clustering Model

Each cluster can be formed whether by moving road users (e.g., a combination of moving vehicles) or by a set of non-moving users (e.g., a group of parked vehicles plus those waiting behind a traffic light). The size of clusters can vary from one road user to two or more. Notice that increasing the number of cluster members is a double-edged sword, increasing the computing capabilities and membership dynamicity. Each cluster has a head (CH) and the rest are cluster members (CMs) that communicate over a wireless network. To join a cluster, a vehicle must contact the *CH*. The *credibility* of cluster members is the foundation of cluster services. In a clustering scenario, a credibility threshold allows new members with higher qualifications (e.g., compared to a threshold) to join the cluster. In each cluster, the *CH* is responsible for letting *CMs* leave the cluster or new members join. Also, the *CH* is responsible for adjusting the credibility threshold to make the cluster more stable [29].

V. THE AR-CFL SYSTEM DESIGN

In Figure 2, we illustrate the main components of AR-CFL. The learning process for object detection is performed on two levels. Firstly, when a user issues a *task* to the central server, a global model is generated to be trained online using vehicles' collected data. For example, if the *task* aims to detect *object 1* in the use case 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 how many vehicles are required based on this specific task's required computation resources. 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 Vertical Cluster-based Federated Learning (VCFL) because it has a vertical

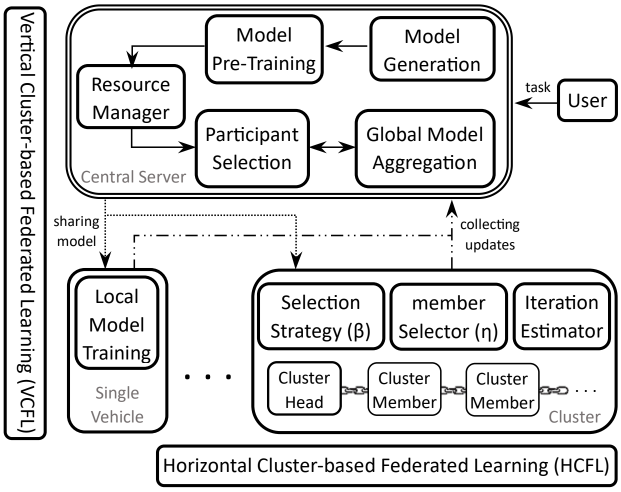


Fig. 2: The AR-CFL System Design, which includes the main components of both levels of federated learning.

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 between the cluster head and members (e.g., in the edge layer), called Horizontal Cluster-based Federated Learning (HCFL). 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.

Although only moving vehicles are involved in cluster formation in traditional clustering scenarios, a substantial intact data collection capacity and computing resources can be explored by incorporating other possibilities. Inspired by Virtual Edge Computing (V-Edge) [30], *virtual clusters* (i.e., V-Cluster) can be formed by connecting not moving road users (e.g., parked cars). The learning process in a V-Cluster is similar to a regular cluster, whereas the communication between V-Cluster members possesses higher stability due to the lack of mobility in these road users. This mainly helps the learning system with less uncertainty in model training, usually caused by unstable or noisy communication channels. Note that since the collected data is not changing (e.g., the same captured images of the road), training the FL model in multiple local iterations does not make sense; thereby, the model will be trained in one local iteration and eventually aggregated by the V-Cluster head.

Each situation requires a particular amount of data and computing resources to be captured from the environment. That's why it is vital to determine the minimum requirements for detecting each situation. We called such metric as the situation's *complexity*. The exact definition of such a parameter is indeed application-dependent and should be determined by domain experts. Moreover, it makes sense to introduce incentives to select clusters over single road users. Therefore, the *Resource Manager* module endeavors to adjust control variables β (i.e., in-cluster participant selection strategy) and η (i.e., number of in-cluster participants) to meet the required Computing Resources (CR) determined by situation *complexity* and update them in each global iteration.

A. VCFL Framework

Due to adding the second level of learning to the conventional FL frameworks, the typical algorithms for client selection and model collection and aggregation should be revised. To be more precise, more detailed synchronization is required between two levels of the FL framework to achieve the expected benefits. In Algorithm 1, we present the procedure of client selection and synchronized learning from the cloud server point of view. In this study, we chose Federate Averaging (FedAvg) [31], a pioneering aggregation approach that achieves better accuracy in previous studies [32], in which a central server (e.g., cloud server in the proposed scenario) hosts the shared global model ω_g, where g stands for the global iteration number in the first level of FL.

At the start of VCFL, the list of available clients with their available computing resources (i.e., $\mathcal{C}$) and the calculated complexity of the target situation (i.e., CR_{s_i}) are initialized. Besides, the central server pre-trains the global model (i.e., ω_0) by utilizing a small set of data gathered from all potential participants, e.g., using one block of data collected by *Cluster 1, Cluster 2,* and *Single Vehicle 1* in the mentioned use case scenario in Section III. Moreover, the participant selection strategy within the cluster is determined here by the central server and sent to the cluster heads so that all clusters perform the selection uniformly (i.e., β). 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 to be involved in the online training procedure is determined here using the η value.

At the start of each global iteration, the central server updates the list of available clients alongside their CR values. The chosen clients are not necessarily the same and can change over time based on availability and CR values. The participant selection method in the first level of learning is the main difference between the proposed method and the typical FedAvg. In more detail, while FedAvg chooses random clients, this approach selects as many cluster participants as possible. The procedure is designed to opt for clients with the highest CR values (i.e., $CR_{c_{max}}$), as clusters usually have higher computing resources. It will continue selecting clients until their computing summation (ii.e., R_g) reaches the threshold required for detecting the situation s_i. Once the selection is finished, each chosen participant receives the global model ω_g and replaces its current local model ω_g^c.

In the next step, depending on the client type, two procedures can be deployed to train the model. Suppose the client is of a type of cluster (normal or virtual). In that case, the second level of FL is called to train the model within the cluster by delivering the global model alongside the β strategy with η number of cluster members (Refer to Algorithm 2). In the end, the cluster head will upload the aggregated update to the central server. Otherwise, if the cluster consists of 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 will generate the updated local model ω_{g+1}^c, which will be 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 ω_{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_σ). This way, more importance is given to the cluster updates, which were trained with more local training iterations.

Algorithm 1 VCFL Procedure

Initialization:
  $\mathcal{C} \leftarrow \{[c_1, CR_{c_1}], ..., [c_n, CR_{c_n}]\}$; ▷ clients
  $CR_{s_i} \leftarrow Complexity(s_i)$; ▷ situation s_i
  Pre-trained ω_0;
  $\beta \leftarrow$ 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
  $E_\sigma \leftarrow 0$; ▷ Sum of Client's Weights
  $C_g \leftarrow \varnothing$; ▷ Selected Clients
  while $R_g \leq CR_{s_i}$ **do**
    $CR_{c_{max}} \leftarrow \max_{i \in \mathcal{C}} CR_i$;
    $C_g \leftarrow C_g \cup c_{max}$;
    $R_g \leftarrow R_g + CR_{c_{max}}$
    $\mathcal{C} \leftarrow \mathcal{C} - [c_{max}, CR_{c_{max}}]$;
  Distribute ω_g to clients in C_g;
  for client $c \in C_g$ **do** ▷ In Parallel
    $\omega_g^c \leftarrow \omega_g$;
    if c is a cluster **then**
      $\omega_{g+1}^c \leftarrow$ HCFL(ω_g^c, β, η); ▷ Algorithm 2
    else
      $P_c \leftarrow$ batches of size B;
      $E_c \leftarrow |P_c|$;
      for partition $p \in P_c$ **do**
        $\omega_g^c \leftarrow$ LocalTraining (ω_g^c, p);
      $\omega_{g+1}^c \leftarrow \omega_g^c$;
  $E_\sigma \leftarrow \sum_{c \in C_g} E_c$;
  $\omega_{g+1} \leftarrow \sum_{c \in C_g} (E_c / E_\sigma) \times \omega_{g+1}^c$;

B. HCFL Framework

As mentioned earlier, the advantages of cluster-based FL are twofold: (a) cluster members attempt to reach a higher level of perception capability for a trained model by performing more iterations than a single vehicle, and (b) fewer data will be exchanged between clients and the central server due to short distance communication within the cluster. The second level of FL is illustrated in Algorithm 2. Once the cluster head receives the global model and values for β and η, it calculates the number of local iterations (i.e., k) required to train the global model,

Algorithm 2 HCFL Procedure

Initialization:
$Cluster \leftarrow \{[m_1, CR_{m_1}], ..., [m_n, CR_{m_n}]\}$;
$CR_\sigma \leftarrow \sum_{m_i \in Cluster} CR_{m_i}$;
$\omega_0^{Cluster} \leftarrow \omega_g^c$;
$k \leftarrow Calculate\_Iterations(CR_\sigma, \beta, \eta)$;
for local iteration l = 0, 1, ..., k **do**
  if $\beta == Full\_Aggregation$ **then**
    $C_l \leftarrow Cluster$;
  else if $\beta == Random$ **then**
    for $i \leftarrow 1$ to η **do**
      $c_i \leftarrow Random(Cluster)$;
      $C_l \leftarrow c_i$;
      $Cluster \leftarrow Cluster - c_i$;
  else if $\beta == MaxLabel$ **then**
    for $i \leftarrow 1$ to η **do**
      $c_i \leftarrow \max_{m_j \in Cluster} LabelCount_{m_j}$;
      $C_l \leftarrow c_i$;
      $Cluster \leftarrow Cluster - c_i$;
  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 ($\omega_l^{c'}$);
    $\Omega \leftarrow \Omega \cup \omega_{l+1}^{c'}$; ▷ Set of Collected Updates
  $\omega_{l+1}^{Cluster} \leftarrow Aggregate(\Omega)$;
return $\omega_k^{Cluster}$;

performed by *Iteration Estimator* module. Due to more effortless synchronization in each round of VCFL, the cluster head tends to minimize the value for k, which helps optimize the convergence time. Although the cluster head in HCFL acts as the central server for model aggregation and distribution, it can also be selected as a participant to train the model with its local data. This aims to avoid wasting the cluster head's resources and help improve the quality of the updated model since the cluster head will provide the captured data and computing resources for the FL approach.

In each local training iteration, the cluster head opts for the set of participants according to the in-cluster participant selection strategy (i.e., β). There are three selection strategies for selecting in-cluster participants β: *Full Aggregation*, *Random*, and *MaxLabel*. When $\beta = FullAggregation$, all cluster members would participate, and models are aggregated at the cluster head. On the other hand, when $\beta = Random$, FL clients are randomly selected in each iteration. Finally, when $\beta = MaxLabels$, cluster members with the most labels (data-rich) are selected (See Figure 3).

In both *Random,* and *MaxLabel* strategies, η specifies how many members will be involved in the local training within each cluster. The central server will determine η based on various conditions (e.g., situation's complexity, 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 $\beta = MaxLabels$, two cluster members with the largest label count in that specific cluster are selected. In the case

(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).

of *FullAggregation* setup, η equals the total number of vehicles in the cluster.

In the next step, each selected member trains the model with fresh local 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 central server as a result of this round of global training procedure (i.e., VCFL).

C. Handling The Limited Storage Challenge

In our research, we introduce a novel aspect by taking into account the constrained storage capacity of the participating vehicles. Data accumulated by a vehicle during a single iteration is utilized to train the model, provided the vehicle is selected to contribute to the model training process. Subsequently, the vehicle purges the utilized data, making room for the acquisition of fresh data. This approach contrasts conventional Federated Learning (FL) literature, as vehicles rely exclusively on their locally stored data within each iteration, eliminating the practice of data re-usage across iterations. 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 β and η, 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.

VI. EVALUATION

In this section, we outline the objectives of our evaluation, which aims to assess the impact of clustering in the context of online learning efficiency and communication overhead, compared with both centralized and classical federated learning approaches. The evaluation encompasses the following aspects:

1) *Effect of Varying Traffic Density:* We analyze how varying traffic density affects the overall system performance.
2) *Improvement in Online Learning Efficiency:* We investigate how clustering enhances the efficiency of online learning in terms of communication overhead and application-related performance (perception capability), as compared to a centralized learning approach.
3) *Enhanced Online Federated Learning Efficiency:* We assess how clustering influences the efficiency of online FL, with a focus on communication overhead and perception capability, in comparison to the classical federated learning approach.
4) *Impact of Cluster Member Selection Strategy:* We explore the influence of various selection strategies (i.e., β), the number of selected cluster members (i.e., η), and the number of clusters on system performance.

Our evaluation considers several key factors:

Limited Vehicle Storage: Throughout our evaluation, we consider the constraint of limited storage on the vehicles and the importance of collecting fresh data over successive iterations.

Non-IID Data: We conduct the evaluation on non-iid data, with a clear characterization of data heterogeneity across system members.

Environmental Considerations: The evaluation is carried out under various environmental conditions to assess the robustness and adaptability of the clustering approach.

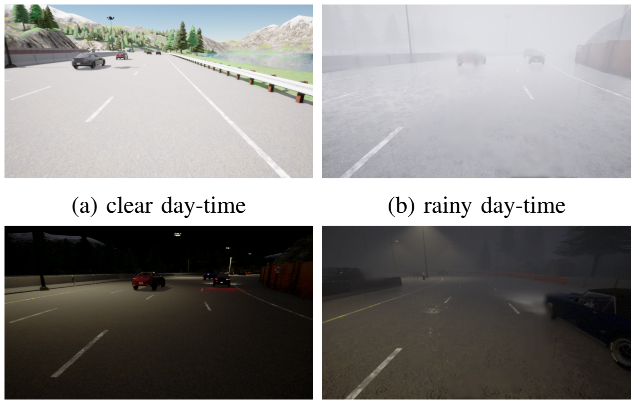
Communication Assumptions: We assume that communication between vehicles within the same cluster is easier to establish and less costly than communicating with edge or cloud servers (long-range communication). This assumption aligns with our hypothesis that reduced long-range communication requirements lead to improved network efficiency.

By systematically addressing these evaluation goals and considerations, we aim to provide a comprehensive assessment of the benefits and performance implications of clustering in the context of online learning in vehicular networks.

A. Evaluation Scenario and Experimental Setup

We detail the evaluation scenario and experimental setup for assessing our approach, which focuses on training a car detection model using image data from participating vehicles. Our goal is to measure the training efficiency compared to baselines. To meet the requirement of having image data from multiple vehicles in similar conditions, we created a synthetic dataset using the Carla simulator [33]. The experiments employed the *Yolo8n* model [34] on a Linux server with an NVidia RTX3090 Ti GPU.

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



(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.

2) General Setup Variables: In our general setup, we maintain a constant total of 12 participating vehicles (N_v) for Federated Learning (FL) model training. We vary traffic density (α) with values of 30, 50, and 100, where $\alpha = 50$ indicates the presence of 50 vehicles. These vehicles are distinct from the 12 data collector vehicles participating in FL model training. Additionally, we explore clustering scenarios, varying the number of clusters (N_{cls}). For $N_{cls} = 2$, each cluster comprises 6 vehicles, while $N_{cls} = 4$ results in clusters of 3 vehicles each. These variables significantly shape our experimental design and help assess the impact of clustering on system performance.

3) Baselines: We benchmark our experimental results against two baseline methods:

Centralized: This baseline method represents the ideal Oracle case where all data is stored centrally. To ensure a fair and controlled comparison, we harmonize the number of iterations used in the Federated Learning setup with the data splits of all vehicles for the specific iteration.

ClassicalFL: In this case, no clustering is considered, and all Federated Learning (FL) clients are situated at the same level. This approach provides a reference point for evaluating the impact of clustering on FL performance.

4) Federated Learning Hyper-parameters: In our experimental setup, we carefully tuned the hyper-parameters for Federated Learning. We chose the total number of global iterations as $E_g = 50$. Upon receiving the model from the server, 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 $batch\_size = 16$. We established the learning rate parameters with $lr0$ and lrf, both configured at their default values of $lr0 = lrf = 0.01$, and we employed $workers = 2$. Additionally, we selected $optimizer = auto$ while maintaining default values for all other model training and validation parameters [35].

B. Evaluation Metrics

To assess the performance of the different approaches, we distinguish between three main categories of metrics, as follows:

1) Perception Capability: In evaluating the performance of object detection models, two key metrics are often used:

- Mean Average Precision (mAP): mAP is a widely used metric for object detection because it considers precision and recall across multiple object classes [36]. mAP is particularly valuable because it considers the object detection capability at different confidence score thresholds, making it a robust evaluation metric. In our evaluation, we considered mAP at various thresholds, such as mAP50, mAP75, and mAP50-95.
- F1 Score: We use the F1 score as a supportive metric to measure the detection performance of the trained model.

2) Training Time: In reference to the total training time denoted as tr_t, 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.

3) Volume of the Exchanged Data: We define e_d to measure the size of the exchanged data while neglecting the generated traffic to select the participating clients in the clustering setups. In addition, we omitted all the other CPM loads for simplicity. In the case of *Centralized* setup, e_d is calculated by measuring the size of the data (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 in our simulation $N_v = 12$ sake of simplicity , 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*. In *ClassicalFL*, 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 l, each selected vehicle sends the model back to the server. Thus, the final formula to calculate e_d in this case is as follows:

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

where $model_s$ indicates the model size.

Finally, we consider two-level aggregation to compute data exchange volumes in clustering setups. First, long-distance communication between the server and cluster heads is crucial. By minimizing data exchange in this costly and slow process, overall efficiency improves. The bandwidth cost, denoted as ed_l, is computed as

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

replacing N_n with N_{cls} (number of clusters).

Second, short-distance communication between cluster heads and members is faster and less costly. The bandwidth cost, denoted as ed_s, is calculated as

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

Depending on client selection methods (β and η), vehicles are chosen from all clusters. For *FullAggregation*, η equals the total cluster members minus one. Actual e_d values for *FullAggregation*, *Random*, and *MaxLabels* are detailed in Table III.

C. Data Generation

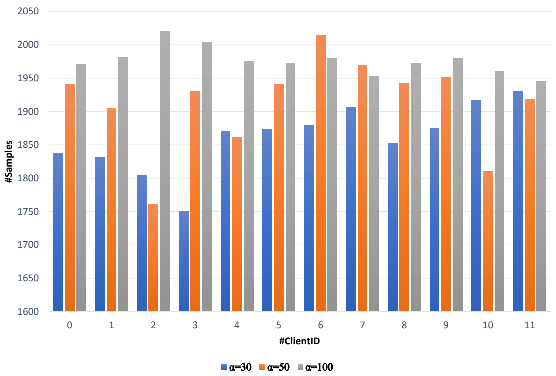
We used the Carla simulator [33] to generate training and validation data, building upon [37] for concurrent image and ground truth data generation from multiple vehicles. As part of ongoing support for researchers, we're developing data-generating files with a web-based interface. Simulations used the pre-built Map-Town04 [38], and data for each situation was uniformly generated. Notably, we've created our dataset and plan to share both the dataset and source code for data generation in future work.

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

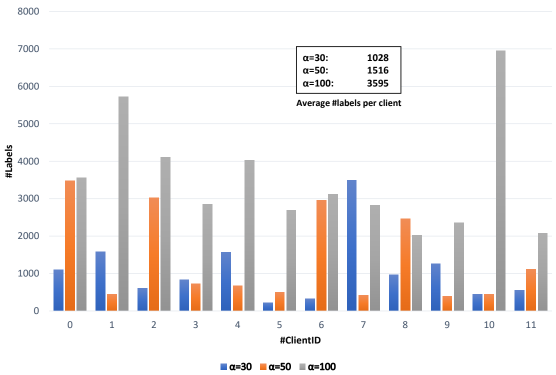
Figure 5b depicts label (car bounding box) distribution across the clients' samples for different α values. Higher α values result in more total labels distributed among clients' samples. The figure illustrates the total number of labels per client across all iterations. Importantly, it does not necessarily indicate that a client with a high total label count has a consistently high label count in all individual global iterations g. On the other hand, the data sample distribution remains consistent among clients within each global iteration g, minimizing data quantity skew. However, a noticeable label distribution imbalance across clients underscores our engagement with non-IID data handling and illustrates data heterogeneity across clients [39], [40].

D. Results and Discussion

We discuss the experimental results to investigate the key factors influencing system performance. We analyze the impact of traffic density fluctuations, explore how clustering improves online learning efficiency, assess the influence on online Federated Learning efficiency, and examine the effects of cluster member selection strategies.

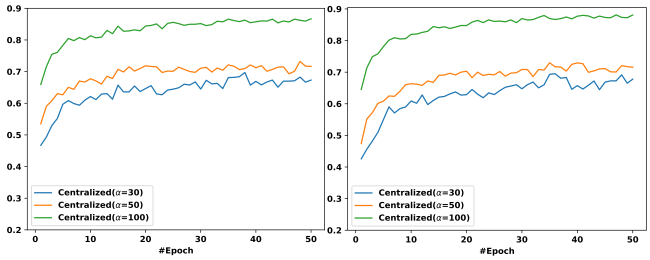


(a) Sample distribution over clients (vehicles).



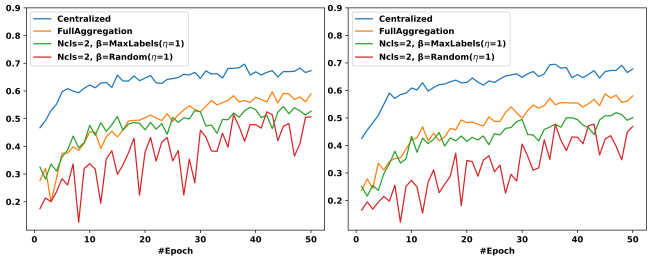
(b) Label distribution over clients (vehicles).

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



(a) F1 (b) mAP50

Fig. 6: The perception capability 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.



(a) F1 (b) mAP50

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

1) Analyzing Traffic Density Impact on Performance: We explored the influence of traffic density on system performance, evaluating under $\alpha = 30, 50, 100$. Figure 6 illustrates some perception capability metrics in the *Centralized* approach, revealing an evident trend: as traffic density rises, there's more consistent model performance and an overall enhancement in the perception capability. 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 perception capability without influencing training time or data exchange volume. These factors depend solely on sample size, not characteristics within the samples. These insights extend beyond the *Centralized* approach, as demonstrated in Tables I, II, and III.

2) Influence of Clustering vs. Centralized Approach on Online Learning Efficiency: In examining the impact of clustering on online learning efficiency versus centralized learning, we emphasize communication overhead and application-related performance. Figures 7 and 8 reveal that the *Centralized* approach consistently outperforms clustering methodologies in perception capability. Despite this, the gap remains constant across different α values.

The true advantage of clustering emerges in reduced training time (tr_t). Clustering approaches demonstrate an impressive 52% decrease in training time compared to *Centralized* ($tr_t = 207 minutes$ for *Centralized* vs. $tr_t = 101 minutes$ for all *Clustering* setups). Moreover, considering maximum exchanged data volume (e.g., *MaxLabels* with $\eta = 5$), clustering setups exhibit a notable 30% reduction compared to *Centralized*. This reduced data exchange is a pivotal factor contributing to the overall efficiency of the clustering-based online learning paradigm. Refer to Tables I and III for comprehensive quantitative results supporting our findings.

3) Influence of Clustering vs. ClassicalFL Approach on Online Learning Efficiency: We analyze how clustering impacts online learning efficiency, emphasizing communication overhead and application-related performance compared to the classical federated learning approach.

Figures 9 and 10 offer nuanced insights into online learning efficiency. *FullAggregation* and *MaxLabels* clustering strategies outperform traditional *ClassicalFL* in perception capability. *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 (ed_s) compared to *ClassicalFL*.

MaxLabels surpasses *ClassicalFL* by selecting clients within each cluster with the maximum labels per iteration, enhancing the perception capability and convergence. Yet, clustering introduces additional communication overhead, evident in ed_s values exclusive to clustering setups (Table

TABLE I: Performance Metrics for the *Centralized* approach across three distinct traffic densities $\alpha = 30, 50, 100$. Data gathered from all clients is used. The training time is $tr_t = 207Minutes$.

α	Perception Capability Metrics				e_d (MB)
	F1	mAP50	mAP75	mAP50-95	
$\alpha = 30$	0.673	0.677	0.537	0.452	10488
$\alpha = 50$	0.716	0.715	0.545	0.475	10786
$\alpha = 100$	0.866	0.880	0.766	0.672	11146

TABLE II: Performance Metrics for *ClassicalFL* method with different N_n values. $N_n = 12$ represents *FullAggregation* and provide similar perception capability in both *ClassicalFL* and *Clustering* approaches. The training time is $tr_t = 101Minutes$, excluding the time spent on model exchange and selecting participating clients.

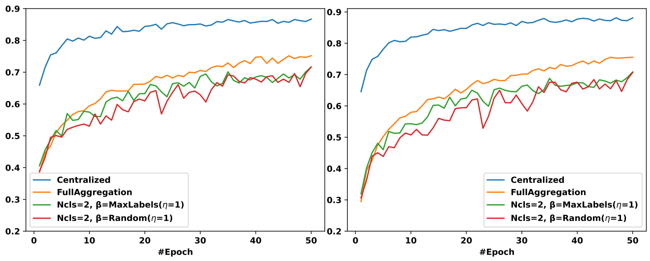
α	N_n	Perception Capability Metrics				e_d (MB)
		F1	mAP50	mAP75	mAP50-95	
$\alpha = 30$	2	0.512	0.466	0.228	0.247	1240
	4	0.540	0.529	0.317	0.299	2480
	12	0.589	0.579	0.359	0.343	7440
$\alpha = 50$	2	0.512	0.484	0.206	0.239	1240
	4	0.570	0.530	0.259	0.279	2480
	12	0.629	0.626	0.381	0.356	7440
$\alpha = 100$	2	0.673	0.663	0.444	0.406	1240
	4	0.723	0.724	0.523	0.465	2480
	12	0.751	0.755	0.552	0.491	7440

III). The *Random* clustering setup with $\eta = 1$ shows comparable performance to *ClassicalFL*, randomly selecting clients in each iteration. With increased η (e.g., $\eta = 3$), *Random* clustering outperforms *ClassicalFL* in perception capability due to more participating clients. Considering only model training and aggregation time and assuming simultaneous training in clusters, both *Clustering* setups and *ClassicalFL* have a training time of $tr_t = 101minutes$. For detailed results, refer to Tables III and II.

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

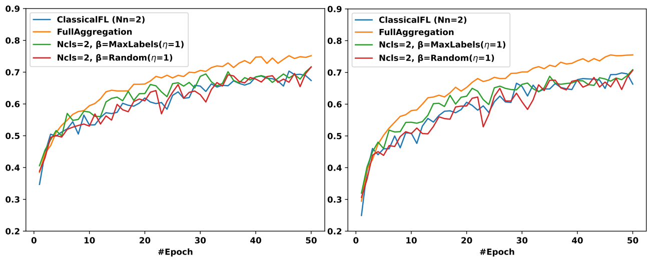
Varying Cluster Numbers (N_{cls}): We conducted a comprehensive analysis to assess the influence of different numbers of clusters, denoted as N_{cls}, on various aspects of our proposed system. Our findings, as presented in Table III, reveal compelling insights into the system's behavior.

Firstly, we observe that the training time remains consistent $tr_t = 101Minutes$ irrespective of changes in the number of clusters N_{cls}. Analyzing the communication overhead, we observe an increase in the long-range communication overhead, denoted as ed_l, corresponding to the augmented 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 ed_s exhibits an upward trend with an increased number of clusters. This trend indicates



(a) F1 (b) mAP50

Fig. 8: Comparing the perception capability between the *Centralized* approach and selected *Clustering* approaches under a traffic density of $\alpha = 100$.



(a) F1 (b) mAP50

Fig. 9: Comparing the perception capability between the *ClassicalFL* approach and selected *Clustering* approaches under a traffic density of $\alpha = 100$ with one selected client at each cluster ($\eta = 1$).

a broader engagement of cluster nodes in online learning.

Turning our attention to perception capability, as depicted in Figure 11, we found that when $\beta = FullAggregation$, the perception capability 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 $\beta = FullAggregation$, as detailed in Table II. Contrastingly, when β takes values of either $\beta = Random$ or $\beta = MaxLabels$, perception capability 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 relationship results in an enhanced perception capability with an increased number of clusters.

Different Selection Strategy (β) with Varying (η): In our study, the constant number of data collection vehicles ($N_v = 12$) limited our ability to increase η values beyond $\eta = 2$ for $N_{cls} = 4$. However, for $N_{cls} = 2$, we conducted experiments with varying $\eta = 1, .., 5$.

Figures 12, 13 illustrate that, when $\beta = Random$, increasing η slightly enhances perception capability but consistently falls short of the perception capability achieved

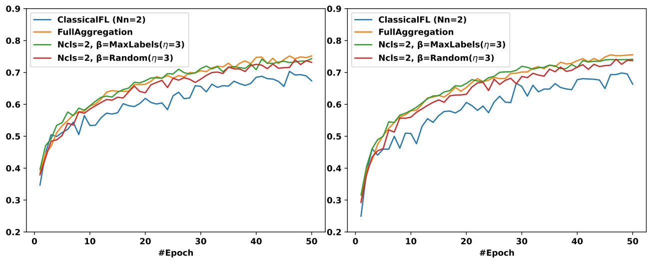
TABLE III: Performance comparison of various *Clustering* approaches across three traffic densities ($\alpha = 30, 50, 100$) and different cluster counts ($N_{cls} = 2, 4$). The training time is $tr_t = 101 Minutes$ for all approaches, excluding the time spent on model exchange and the selection process for participating clients.

α	N_{cls}	β	Perception Capability Metrics				e_d (MB)	
			F1	mAP50	mAP75	mAP50-95	ed_l	ed_s
$\alpha = 30$	2	*FullAggregation*	0.589	0.579	0.359	0.343	1240	6200
		Random ($\eta = 1$)	0.506	0.469	0.242	0.248	1240	1240
		Random ($\eta = 2$)	0.561	0.545	0.332	0.314	1240	2480
		Random ($\eta = 3$)	0.566	0.56	0.339	0.326	1240	3720
		Random ($\eta = 4$)	0.605	0.58	0.348	0.336	1240	4960
		Random ($\eta = 5$)	0.583	0.579	0.353	0.343	1240	6200
		MaxLabels ($\eta = 1$)	0.527	0.501	0.308	0.289	1240	1240
		MaxLabels ($\eta = 2$)	**0.596**	0.574	**0.362**	0.34	1240	2480
		MaxLabels ($\eta = 3$)	0.583	0.572	**0.359**	0.34	1240	3720
		MaxLabels ($\eta = 4$)	**0.599**	**0.587**	**0.365**	**0.351**	1240	4960
		MaxLabels ($\eta = 5$)	**0.593**	**0.585**	0.358	**0.349**	1240	6200
	4	*FullAggregation*	0.589	0.579	0.359	0.343	2480	4960
		Random ($\eta = 1$)	0.558	0.549	0.345	0.327	2480	2480
		Random ($\eta = 2$)	0.588	0.578	0.348	0.333	2480	4960
		MaxLabels ($\eta = 1$)	0.574	0.567	0.331	0.323	2480	2480
		MaxLabels ($\eta = 2$)	**0.598**	**0.587**	**0.368**	**0.348**	2480	4960
$\alpha = 50$	2	*FullAggregation*	0.629	0.626	0.381	0.356	1240	6200
		Random ($\eta = 1$)	0.529	0.488	0.212	0.249	1240	1240
		Random ($\eta = 2$)	0.578	0.566	0.319	0.313	1240	2480
		Random ($\eta = 3$)	0.588	0.565	0.315	0.317	1240	3720
		Random ($\eta = 4$)	0.625	0.61	0.359	0.342	1240	4960
		Random ($\eta = 5$)	0.616	0.609	0.378	0.354	1240	6200
		MaxLabels ($\eta = 1$)	0.586	0.560	0.365	0.340	1240	1240
		MaxLabels ($\eta = 2$)	0.619	0.605	**0.412**	**0.366**	1240	2480
		MaxLabels ($\eta = 3$)	0.628	**0.627**	**0.426**	**0.374**	1240	3720
		MaxLabels ($\eta = 4$)	0.621	0.619	**0.406**	**0.365**	1240	4960
		MaxLabels ($\eta = 5$)	0.612	0.613	0.366	0.343	1240	6200
	4	*FullAggregation*	0.589	0.579	0.359	0.343	2480	4960
		Random ($\eta = 1$)	0.552	0.530	0.309	0.291	2480	2480
		Random ($\eta = 2$)	0.602	0.604	0.339	0.333	2480	4960
		MaxLabels ($\eta = 1$)	**0.594**	0.569	**0.378**	**0.346**	2480	2480
		MaxLabels ($\eta = 2$)	**0.619**	**0.617**	**0.361**	**0.344**	2480	4960
$\alpha = 100$	2	*FullAggregation*	0.751	0.755	0.552	0.491	1240	6200
		Random ($\eta = 1$)	0.716	0.706	0.483	0.439	1240	1240
		Random ($\eta = 2$)	0.723	0.718	0.540	0.456	1240	2480
		Random ($\eta = 3$)	0.731	0.737	0.528	0.474	1240	3720
		Random ($\eta = 4$)	0.734	0.742	0.536	0.481	1240	4960
		Random ($\eta = 5$)	0.741	0.746	**0.54**	**0.479**	1240	6200
		MaxLabels ($\eta = 1$)	0.716	0.708	0.498	0.450	1240	1240
		MaxLabels ($\eta = 2$)	0.73	0.74	0.551	0.485	1240	2480
		MaxLabels ($\eta = 3$)	0.743	0.741	**0.557**	0.49	1240	3720
		MaxLabels ($\eta = 4$)	0.745	0.745	**0.56**	**0.493**	1240	4960
		MaxLabels ($\eta = 5$)	**0.765**	**0.759**	**0.555**	**0.497**	1240	6200
	4	*FullAggregation*	0.751	0.755	0.552	0.491	2480	4960
		Random ($\eta = 1$)	0.725	0.725	0.522	0.458	2480	2480
		Random ($\eta = 2$)	0.745	0.754	0.537	0.488	2480	4960
		MaxLabels ($\eta = 1$)	0.727	0.743	0.534	0.479	2480	2480
		MaxLabels ($\eta = 2$)	0.744	0.75	**0.559**	**0.491**	2480	4960

with $\beta = FullAggregation$. Conversely, with $\beta = MaxLabels$, increasing η notably improves perception performance. Furthermore, we observed that with 16-25% fewer participating nodes, $\beta = MaxLabels$ outperforms $\beta = FullAggregation$. In addition, $\beta = MaxLabels$ requires approximately 17-33% less long-range communication for the same outcome. Table III reveals an interesting trend: while increasing η generally boosts perception capability, a threshold exists beyond which the positive effect diminishes. For instance, in the case of $\alpha = 30$, $N_{cls} = 2$, perception capability increases with growing η until $\eta = 4$, which starts to decline. This decline is attributed to $\beta = MaxLabels$, where nodes with fewer labels are excluded from online learning, preventing potential negative effects. However, when η surpasses a certain threshold, nodes with minimal labels are included, leading to decreased perception capability. 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 perception capability drops, compared to the continuous increase with $\alpha = 100$, $N_{cls} = 2$.

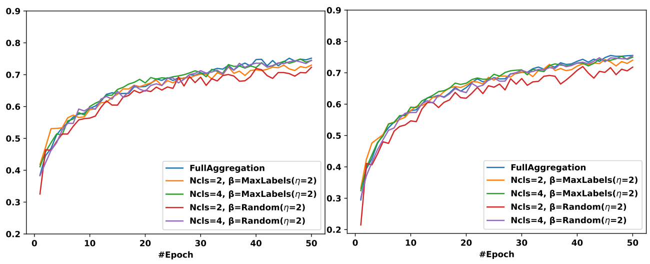
E. Limitations of the Study

We exchanged the entire model due to the detection model's modest size (6.2 MB) during our experiments. However, practical perception models may be larger, necessitating model compression for enhanced efficiency. A limitation involves the need for image data captured



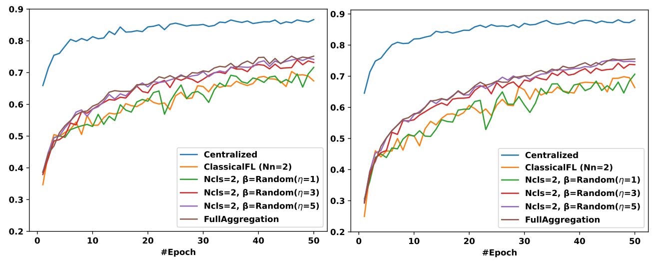
(a) F1 (b) mAP50

Fig. 10: Comparing the perception capability between the *ClassicalFL* approach and selected *Clustering* approaches under a traffic density of $\alpha = 100$ with three selected client at each cluster ($\eta = 3$).



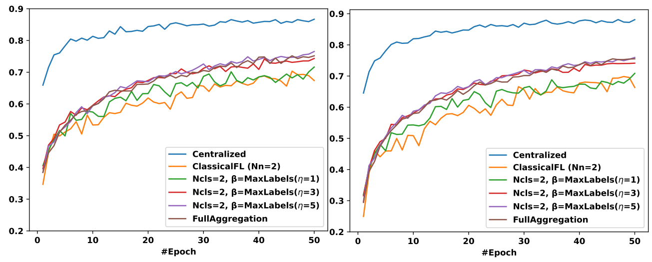
(a) F1 (b) mAP50

Fig. 11: The perception capability 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}.



(a) F1 (b) mAP50

Fig. 12: Comparing the perception capability between the *FullAggregation* and *Random* clustering approaches under a traffic density of $\alpha = 100$, with a varying number of selected clients at each cluster ($\eta = 1, .., 5$), and two clusters ($N_{cls} = 2$).



(a) F1 (b) mAP50

Fig. 13: Comparing the perception capability between the *FullAggregation* and *MaxLabels* clustering approaches under a traffic density of $\alpha = 100$, with a varying number of selected clients at each cluster ($\eta = 1, .., 5$), and two clusters ($N_{cls} = 2$).

by multiple vehicles in close proximity under similar environmental conditions. We addressed this by generating synthetic datasets using the Carla simulator [33], but real-world data would offer a more accurate representation. Notably, we did not incorporate security or privacy-preserving mechanisms in this paper. For a more comprehensive approach, integrating the encryption and privacy-preserving techniques (e.g., differential privacy) is advisable. These aspects are recognized as important directions for future research and refinement of the proposed framework.

VII. Conclusion and Future Work

In this paper, we presented AR-CFL, an innovative Adaptive Resource-aware Clustered Federated Learning framework designed specifically for a thorough examination of 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 car detection models online training on non-IID data, considering various conditions. To achieve this objective, we created three synthetic image datasets representing different traffic densities using the Karla simulator. In contrast to existing literature, we addressed the constraint of limited storage on vehicles by utilizing freshly captured data at each global iteration. Our analysis revealed that increasing the traffic density enhances the perception capability of the train model. We compared the *Clustering* approaches against both *Centralized* and traditional *ClassicalFL* learning approaches under different configurations. Results demonstrate a noteworthy advantage in leveraging FL, showcasing a remarkable 52% reduction in training time compared to the traditional centralized training approach. 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 heightened long-range communication. Notably, certain participant selection strategies, such as *MaxLabels*, demonstrate high perception performance compared to *FullAggregation* approach, with up to a 25% reduction in participating nodes, and 33% less long-range communication.

As a direction for future research, 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 would offer significant benefits.

References

[1] Shuncheng Liu, Yuyang Xia, Xu Chen, Jiandong Xie, Han Su, and Kai Zheng. Impact-aware maneuver decision with enhanced perception for autonomous vehicle. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 3255–3268. IEEE, 2023.

[2] Yuxiao Zhang, Alexander Carballo, Hanting Yang, and Kazuya Takeda. Perception and sensing for autonomous vehicles under adverse weather conditions: A survey. *ISPRS Journal of Photogrammetry and Remote Sensing*, 196:146–177, 2023.

[3] Ahmad Khalil, Tobias Meuser, Yassin Alkhalili, Antonio Fernández Anta, Lukas Staecker, and Ralf Steinmetz. Situational collective perception: Adaptive and efficient collective perception in future vehicular systems. In *International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS'22)*, pages 346–352, 2022.

[4] Quentin Delooz, Alexander Willecke, Keno Garlichs, Andreas-Christian Hagau, Lars Wolf, Alexey Vinel, and Andreas Festag. Analysis and evaluation of information redundancy mitigation for v2x collective perception. *IEEE Access*, 10:47076–47093, 2022.

[5] Majid Lotfian Delouee, Boris Koldehofe, and Viktoriya 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 (DEBS'23)*, page 13–24. ACM, 2023.

[6] Majid Lotfian Delouee, Boris Koldehofe, and Viktoriya Degeler. Towards adaptive quality-aware complex event processing in the internet of things. In *Proceedings of the 18th International Conference on Mobility, Sensing and Networking (MSN'22)*, pages 571–575. IEEE, 2022.

[7] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on non-iid data silos: An experimental study. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 965–978. IEEE, 2022.

[8] Majid Lotfian Delouee, Boris Koldehofe, and Viktoriya 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 (DEBS'23)*, page 185–186. ACM, 2023.

[9] Linlin You, Sheng Liu, Yi Chang, and Chau Yuen. A triple-step asynchronous federated learning mechanism for client activation, interaction optimization, and aggregation enhancement. *IEEE Internet of Things Journal*, 9(23):24199–24211, 2022.

[10] Min Chen, Yang Xu, Hongli Xu, and Liusheng Huang. Enhancing decentralized federated learning for non-iid data on heterogeneous devices. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 2289–2302. IEEE, 2023.

[11] Yansheng Wang, Yongxin Tong, Zimu Zhou, Ruisheng Zhang, Sinno Jialin Pan, Lixin Fan, and Qiang Yang. Distribution-regularized federated learning on non-iid data. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 2113–2125. IEEE, 2023.

[12] Yaxin Luopan, Rui Han, Qinglong Zhang, Chi Harold Liu, Guoren Wang, and Lydia Y Chen. Fedknow: Federated continual learning with signature task knowledge integration at edge. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 341–354. IEEE, 2023.

[13] Hyungbin Kim, Byungchan Kim, Yongho Kim, Cheolwoo You, and Hyunhee Park. K-fl: Kalman filter-based clustering federated learning method. *IEEE Access*, 2023.

[14] Zaobo He, Lintao Wang, and Zhipeng Cai. Clustered federated learning with adaptive local differential privacy on heterogeneous iot data. *IEEE Internet of Things Journal*, 2023.

[15] Deepthi Jallepalli, Navya Chennagiri Ravikumar, Poojitha Vurtur Badarinath, Shravya Uchil, and Mahima Agumbe Suresh. Federated learning for object detection in autonomous vehicles. In *2021 IEEE Seventh International Conference on Big Data Computing Service and Applications (BigDataService)*, pages 107–114. IEEE, 2021.

[16] Gaith Rjoub, Omar Abdel Wahab, Jamal Bentahar, and Ahmed Saleh Bataineh. Improving autonomous vehicles safety in snow weather using federated yolo cnn learning. In *International Conference on Mobile Web and Intelligent Information Systems*, pages 121–134. Springer, 2021.

[17] Tianyue Zheng, Ang Li, Zhe Chen, Hongbo Wang, and Jun Luo. Autofed: Heterogeneity-aware federated multimodal learning for robust autonomous driving. *arXiv preprint arXiv:2302.08646*, 2023.

[18] Shuai Wang, Chengyang Li, Derrick Wing Kwan Ng, Yonina C Eldar, H Vincent Poor, Qi Hao, and Chengzhong Xu. Federated deep learning meets autonomous vehicle perception: Design and verification. *IEEE network*, 2022.

[19] Yiyue Chen, Chianing Wang, and Baekgyu Kim. Federated learning with infrastructure resource limitations in vehicular object detection. In *2021 IEEE/ACM Symposium on Edge Computing (SEC)*, pages 366–370. IEEE, 2021.

[20] Xiaokang Zhou, Wei Liang, Jinhua She, Zheng Yan, I Kevin, and Kai Wang. Two-layer federated learning with heterogeneous model aggregation for 6g supported internet of vehicles. *IEEE Transactions on Vehicular Technology*, 70(6):5308–5317, 2021.

[21] Ahmet M Elbir, Sinem Coleri, Anastasios K Papazafeiropoulos, Pandelis Kourtessis, and Symeon Chatzinotas. A hybrid architecture for federated and centralized learning. *IEEE Transactions on Cognitive Communications and Networking*, 8(3):1529–1542, 2022.

[22] Renhao Lu, Weizhe Zhang, Yan Wang, Qiong Li, Xiaoxiong Zhong, Hongwei Yang, and Desheng Wang. Auction-based cluster federated learning in mobile edge computing systems. *IEEE Transactions on Parallel and Distributed Systems*, 34(4):1145–1158, 2023.

[23] Beibei Li, Yukun Jiang, Wenbin Sun, Weina Niu, and Peiran Wang. Fedvanet: Efficient federated learning with non-iid data for vehicular ad hoc networks. In *2021 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2021.

[24] Moming Duan, Duo Liu, Xinyuan Ji, Yu Wu, Liang Liang, Xianzhang Chen, Yujuan Tan, and Ao Ren. Flexible clustered federated learning for client-level data distribution shift. *IEEE Transactions on Parallel and Distributed Systems*, 33(11):2661–2674, 2021.

[25] Afaf Taik, Zoubeir Mlika, and Soumaya Cherkaoui. Clustered vehicular federated learning: Process and optimization. *IEEE Transactions on Intelligent Transportation Systems*, 23(12):25371–25383, 2022.

[26] Ryuichi Fukatsu and Kei Sakaguchi. Millimeter-wave v2v communications with cooperative perception for automated driving. In *Proceeding of the 89th vehicular technology conference (VTC'19-Spring)*, pages 1–5. IEEE, 2019.

[27] Christoph Pilz, Andrea Ulbel, and Gerald Steinbauer-Wagner. The components of cooperative perception-a proposal for future works. In *Proceeding of the International Intelligent Transportation Systems Conference (ITSC'21)*, pages 7–14. IEEE, 2021.

[28] Aaron Yi Ding, Ella Peltonen, Tobias Meuser, Atakan Aral, Christian Becker, Schahram Dustdar, Thomas Hiessl, Dieter Kranzlmüller, Madhusanka Liyanage, Setareh Maghsudi, et al. Roadmap for edge ai: A dagstuhl perspective, 2022.

[29] Xiaoting Ma, Junhui Zhao, and Yi Gong. Joint scheduling and resource allocation for efficiency-oriented distributed learning over vehicle platooning networks. *IEEE Transactions on Vehicular Technology*, 70(10):10894–10908, 2021.

[30] Falko Dressler, Carla Fabiana Chiasserini, Frank H.P. Fitzek, Holger Karl, Renato Lo Cigno, Antonio Capone, Claudio Casetti, Francesco Malandrino, Vincenzo Mancuso, Florian Klingler, and Gianluca Rizzo. V-edge: Virtual edge computing as an enabler for novel microservices and cooperative computing. *IEEE Network*, 36(3):24–31, 2022.

[31] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistic, PMLR 54*, pages 1273–1282, 2017.

[32] Adrian Nilsson, Simon Smith, Gregor Ulm, Emil Gustavsson, and Mats Jirstrand. A performance evaluation of federated learning algorithms. In *Proceedings of the Second Workshop on Distributed Infrastructures for Deep Learning (DIDL'18)*, pages 1—-8. ACM, 2018.

[33] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

[34] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. YOLO by Ultralytics, January 2023.

[35] Ultralytics. Ultralytics documentation: Train mode arguments, 2023.

[36] Rafael Padilla, Sergio L Netto, and Eduardo AB Da Silva. A survey on performance metrics for object-detection algorithms. In *2020 international conference on systems, signals and image processing (IWSSIP)*, pages 237–242. IEEE, 2020.

[37] AveesLab. Carfree: Automatic ground truth generation in carla, 2021. GitHub repository.
[38] CARLA Simulator. Carla documentation - map town04, 2022. Online documentation.
[39] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.
[40] Ahmad Khalil, Aidmar Wainakh, Ephraim Zimmer, Javier Parra-Arnau, Antonio Fernández Anta, Tobias Meuser, Ralf Steinmetz, et al. Label-aware aggregation for improved federated learning. In *IEEE International Conference on Fog and Mobile Edge Computing*, 2023.