Decentralized Federated Learning for Extended Sensing in 6G Connected Vehicles[⋆]

Luca Barbieri[a,*], Stefano Savazzi[b], Mattia Brambilla[c] and Monica Nicoli[c]

[a] *Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Via Ponzio 34/5, 20133, Milano Italy*
[b] *Institute of Electronics, Inform. Eng. and Telecom. (IEIIT), Consiglio Nazionale delle Ricerche, P.za Leonardo da Vinci, 32, 20133, Milano, Italy*
[c] *Department of Management, Economics and Industrial Engineering, Politecnico di Milano, Via Lambruschini, 4/B, 20156, Milano, Italy*

ARTICLE INFO

Keywords:
cooperative sensing, connected automated driving, distributed computing, federated learning, 6G V2X, consensus

ABSTRACT

Research on smart connected vehicles has recently targeted the integration of vehicle-to-everything (V2X) networks with Machine Learning (ML) tools and distributed decision making. Among these convergent paradigms, Federated Learning (FL) allows the vehicles to train a deep ML model collaboratively, by exchanging model parameters (i.e., neural network weights and biases), rather than raw sensor data, via V2X links. Early FL approaches resorted to a server-client architecture, where a Parameter Server (PS) acts as edge device to orchestrate the learning process. Novel FL tools, on the other hand, target fog architectures where the model parameters are mutually shared by vehicles and synchronized in a distributed manner via consensus. These tools do not rely on the PS, but take advantage of low-latency V2X links. In line with this recent research direction, in this paper we investigate distributed FL for augmenting the capability of road user/object classification based on Lidar data. More specifically, we propose a new modular, decentralized approach to FL, referred to as consensus-driven FL (C-FL), suitable for PointNet compliant deep ML architectures and Lidar point cloud processing for road actor classification. The C-FL process is evaluated by simulating a realistic V2X network, based on the Collective Perception Service (CPS), for mutual sharing of the PointNet model parameters. The performance validation considers the impact of dense connectivity, continual learning over heterogeneous training data, convergence time and loss/accuracy tradeoffs. Experimental results indicate that FL complies with the extended sensors use cases for high levels of driving automation, it provides a low-latency training service, compared with existing distributed ML approaches, and outperforms ego learning with minimal bandwidth usage.

1. Introduction

The forthcoming automated mobility ecosystem is expected to wave in cooperative sensing, communication, computing and learning from massive data in real time [1]. In particular, connected automated driving (CAD) functions target cooperative multi-vehicle control and planning strategies [2]. They rely on the extensive use of Distributed Machine Learning (DML) techniques to achieve robust and reliable predictions over a large number of driving-related functions [3]. Most notably, targeted functions include environmental perception and localization [4], path planning and behaviour arbitration [5, 6], traffic flow prediction [7] and motion control [8]. Solving such complex tasks requires big-data-driven training of large-size Machine Learning (ML) models as well as ultra-reliable low-latency vehicle-to-everything (V2X) interactions with road infrastructure (V2I) and other vehicles (V2V) for cooperative sensing/manoeuvring tasks. In DML, networked vehicles are expected to cooperate by sharing data over V2X links to improve safety, efficiency and driving comfort. However, existing DML techniques in this field of application are based on traditional big-data fusion and cloud/edge principles where vehicles mostly act as data producers [9] while their learning capabilities are not exploited.

Over the past few years, Federated Learning (FL) tools [10–13] have been emerging for addressing large-scale distributed training across many interconnected devices, or agents, with enhanced privacy preserving functionalities compared to DML systems. FL paradigms rely on the exchange of locally trained instances of the ML parameters, i.e., the weights and biases of the Neural Networks (NN), rather than sharing raw data as in DML [9]. As opposed to classical big-data fusion approaches, FL makes use of on-device learning functions and an intensive exchange of ML parameters over the network. This is expected to bring significant benefits in future vehicular scenarios, if rooted in 6th Generation (6G) cellular V2X paradigms that can truly support zero-latency and high-data rate communications in dynamic scenarios.

In early vanilla FL implementations [13], a Parameter Server (PS) acts as a central orchestrator of the learning process: it collects the local models from the participating nodes, updates the received models according to some predefined aggregation procedure, and sends back the aggregated (or global) model to devices. In analogy with distributed ledger and fog computing tools, more advanced designs have been recently proposed to let the local model parameters be consensually shared [14] and synchronized

[⋆] This work is partially supported by the funding program POR-FESR 2014-2020 Project BASE5G (Broadband Interfaces and Services for Smart Environments enabled by 5G technologies) no. 1155850.

[*] Corresponding author: Luca Barbieri
Email addresses: luca1.barbieri@polimi.it (L. Barbieri); stefano.savazzi@ieiit.cnr.it (S. Savazzi); mattia.brambilla@polimi.it (M. Brambilla); monica.nicoli@polimi.it (M. Nicoli)
ORCID(s): 0000-0002-8948-8158 (L. Barbieri); 0000-0002-9865-6512 (S. Savazzi); 0000-0001-5442-6507 (M. Brambilla); 0000-0001-7104-7015 (M. Nicoli)

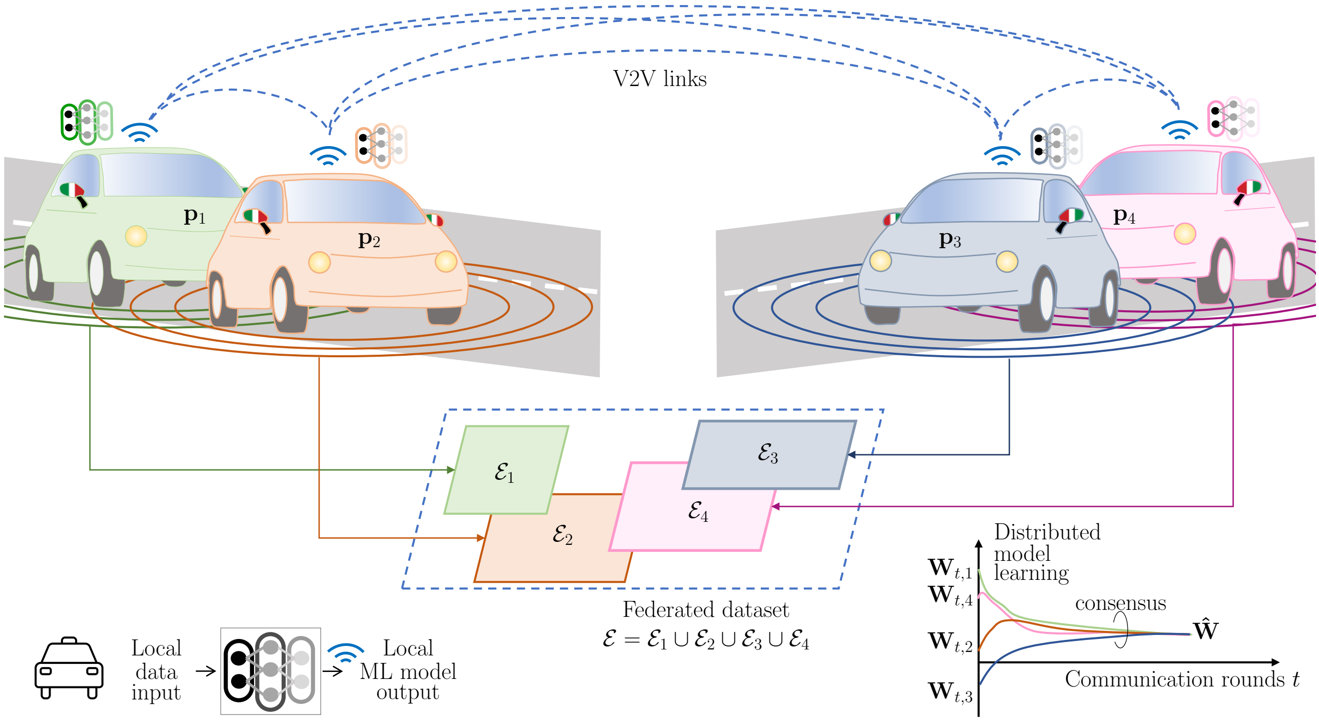


Figure 1: Decentralized FL architecture: vehicles cooperate in the creation of a federated dataset $\mathcal{E}$ comprising the individual dataset $\mathcal{E}_i$ of each vehicle $i \in \mathcal{V}$. Vehicles engage in a distributed ML model optimization by sharing their own parameters $\mathbf{W}_{t,i}$ every communication round t. This allows to reach a consensus on the global model $\hat{\mathbf{W}}$, common to all connected vehicles.

across multiple devices via decentralized, mesh, or device-to-device (D2D) networking, without relying on PS. FL policies well address scalability and privacy shortcomings, although many key problems still need to be solved.

1.1. FL challenges in connected automated vehicles

The main challenges in FL design are device sampling, convergence and statistical heterogeneity [15–18]. For example, device sampling methods allow to optimize the population of devices participating to the learning process, considering data quality, latency and energy tradeoffs. Device selection policies typically target a fixed percentage of devices [13] that are chosen to speed up the learning process [19] or meet eligibility criteria [20]. Reducing convergence time of FL procedures is of paramount importance, especially in mobility scenarios with electric vehicles and latency-critical services. In [21], a gradient-based user association scheme is adopted, while [22, 23] focus on momentum-based optimization and control algorithms to select the best trade-off between local updates and global aggregation. FL setups are characterized by statistical heterogeneity which can highly influence the quality of the trained models. To solve this problem, a variant of Federated Averaging (FedAvg) has been proposed in [24] to restrain local models from diverging from the global one. Other approaches rely on Multi-Task Learning (MTL) [25] or Transfer Leaning (TL) [26] that let devices learn and/or fine-tune their models on their local data.

Centralized, or vanilla FL setups rely on a server/client architecture for carrying out the training process. As the client connections increases, the PS may become a bottleneck of the system, slowing down the entire learning process and becoming unresponsive. On the other hand, decentralized FL techniques bring further advances as they exploit a fully distributed network architecture. Rather than relying on a PS, the cooperating devices exchange the model parameters through peer-to-peer connections and implement a consensus policy. Gossip learning approaches have been proposed in [27], where nodes exchange local models with typically two random neighbors. Consensus-driven policies presented in [11] proposed a consensus based federated averaging tool for model parameters and gradients exchange. Further research studies target analog and digital implementations [28] in fading channels.

FL has recently started to gain attention in connected automated driving applications where vehicles need to handle multi Gbps data streams [29, 30] for cooperative sensing and maneuvering functionalities. Sharing such large volumes of data is unfeasible, but FL allows to reduce the exchange to a more parsimonious set of ML model parameters. Furthermore, it has the advantage of being privacy preserving as it does not require raw sensor data sharing. Even though the application is very relevant due to the standardization and implementation effort made by the 3rd Generation Partnership Project (3GPP) and the 5G Automotive Association (5GAA), few studies have been performed for automotive use cases. In [31], a FL approach is developed for joint power and resource allocation in vehicle-to-vehicle (V2V) communications. A FL-based traffic prediction algorithm is presented in [32], while in [33] a federated transfer Reinforcement Learning (RL) approach is developed for real-time knowledge extraction. Other studies focus on model selection for aggregation [34, 35] or contract-based FL to maximize energy efficiency in electric vehicle networks [36]. Blockchain-based FL algorithms are proposed in [37, 38] for privacy-aware vehicular communications and asynchronous FL deep RL-based node selection. Similar approaches are presented in [39] and [40], where a trusted consensus policy and a blockchain-enabled federated vehicular network are developed, respectively. Finally, [41] presents an initial feasibility study on FL for vehicular networks. The aforementioned methods, however, rely on centralized solutions for model aggregation and/or blockchain technologies leveraging computationally expensive Proof-of-Work (PoW) protocols for verifying model updates that may also utilize Road Side Unit (RSU) or centralized entities. In vehicular scenarios, on the other hand, decentralized solutions are preferred as the connection to the road infrastructure might not be always available and local processing allows to speed up the learning process.

1.2. Contributions and paper organization

In this paper, we investigate the FL technology for the classification of road users or objects (here referred to as road actors) based on Lidar data collected by networked vehicles. A preliminary study on FL system for road actor classification has been performed in [42]. Herein, we extend the previous work by proposing a new cooperative solution based on a fully decentralized FL policy, referred to as consensus-driven FL (C-FL). The main contributions are summarized as follows:

- The proposed C-FL technique targets fully distributed vehicular networks and exploits V2V communications to augment the recognition capability of ego-approaches, while overcoming privacy and flexibility shortcomings of current DML policies. Compared to vanilla FL, the consensus technique makes the distributed learning process more scalable and fault resilient as the number of cooperating vehicles increases. Furthermore, it requires a minimal increase of the computational resources as it is based on weighted averaging.

- Considering FL over deep architectures, typically adopted for solving automated vehicle tasks, a modular approach is proposed, in which the FL process can be enabled across a variable number of ML model layers. The approach is integrated with the C-FL policy and validated using Lidar sensor's readings that are used as input to a Deep Learning (DL) model, namely PointNet [43], which is in charge of inferring the actual type of road actors.

- The C-FL tool is validated over a realistic V2X network that allows the vehicles to cooperate and mutually share the NN parameters. The V2X connections are generated using the trajectories extracted from a microscopic traffic simulator. The FL performance is then assessed over different traffic density conditions, evaluating the impact of the V2X connectivity on the consensus process, the convergence time and the loss/accuracy trade off.

- Concerning the FL implementation into the V2X network protocol, we propose the use of the Collective Perception Message (CPM) service [44] to encapsulate the NN parameters and propagate them through the V2X network. The C-FL framework is also adapted to accommodate 6G-enabled vehicular networks showing how the training process can speed up by leveraging highly reliable and ultra low-latency communications.

Real measurements extracted from a large-scale automotive dataset are used to assess the developed techniques' performance. Experimental results indicate that FL is particularly effective, compared with ego learning approaches, even in more challenging, but practical, settings characterized by heterogeneous data distributions, or non-independent and identically distributed (non-IID). These settings are indeed representative of real-world scenarios where vehicles with outdated, or partially trained models, coexist with highly-automated fully-equipped vehicles and benefit from their cooperation. Ego learning approaches are characterized by low convergence time compared to FL procedures, however, they are not able to provide the required level of accuracy. On the other hand, DML implementations show substantially higher training time compared to the proposed approach due to the raw data fusion at the data center, making them unfeasible in latency-critical vehicular applications.

The paper is organized as follows. Sec. 2 presents the system model employed for characterizing the decentralized FL training process as well as the V2X network. Sec. 3 introduces the proposed C-FL method, a communication-efficient modular approach for model parameters exchange, and the expected key technology enablers in 6G V2X standardization. Sec. 4 describes the overall FL framework, focusing on the vehicular scenario, the dataset, and the PointNet model and its FL adaptation for road actor classification. Finally, Sec. 5 evaluates the proposed C-FL tool as compared with centralized FL, DML and ego learning strategies. Learning loss, accuracy and convergence time are evaluated for different V2X connectivity scenarios as well as for continual learning setups.

2. System model

We consider a vehicular scenario evolving over time t where a set $\mathcal{V} = \{1, \cdots, N_v\}$ of N_v vehicles participate in the FL process. We assume a same number of vehicles simultaneously present at each time instant t, thus N_v is time-invariant. The position of vehicle $i \in \mathcal{V}$ over the 2D space is denoted by $\mathbf{p}_{t,i} = [p_{t,i,x}, p_{t,i,y}]^{\mathrm{T}}$.

By moving along their paths, the vehicles engage in a dynamic cooperation process, where V2V links act as a bridge among vehicles, letting them share information with the nearby ones. We model this process as a time-variant graph $\mathcal{G}_t$, where vehicles $\mathcal{V}_t$ are the nodes and the edges ξ_t are the V2V links. The directed graph is thus defined as $\mathcal{G}_t = \left(\mathcal{V}_t, \xi_t\right)$. For a given vehicle i, the set of its neighbors at time t is indicated by $\mathcal{V}_{t,i} = \{j \in \mathcal{V}, j \neq i : \|\mathbf{p}_{t,i} - \mathbf{p}_{t,j}\| \leq R_c\} \subseteq \mathcal{V}$, being R_c the V2V connection range (here assumed to be equal among all vehicles). An ideal free-space communication is considered between each vehicle pair, without any obstructing objects that prevent the V2V connection.

In the following, we first describe the reference use case scenario targeting augmented sensing for road object classification (Sec. 2.1), then we provide details about the V2X communication architecture and protocol, as well as a description of the NN parameters that are mutually exchanged among vehicles (Sec. 2.2).

2.1. Cooperative sensing for road actor classification

In the proposed setup, the vehicles detect the surrounding road actors (such as pedestrians, traffic cones, barriers, bicycles, cars and buses) by scanning the environment through a Lidar sensor. The sensing range is limited to 70 m, and the spatial accuracy of each point cloud is ±2 cm. The Lidar Field of View (FOV) is 360 deg in azimuth and $[+10\,, -30]$ deg in elevation, while the capture frequency is 20 Hz.

We propose a cooperative sensing methodology where the goal of vehicles is to cooperatively learn a PointNet-based deep ML model for inferring the actual type of road actors using Lidar point clouds. Differently from non-cooperative algorithms that operate locally on each ego vehicle, which can optimize the ML model by relying only on its own dataset $\mathcal{E}_i$, in FL all vehicles aim to optimize their own model by including information from the neighbors. This leads to the creation of an aggregated dataset $\mathcal{E} = \bigcup_{i=1}^{N_v} \mathcal{E}_i$, where each ego dataset $\mathcal{E}_i$ can have different size and/or include a limited set of classes, i.e., $\mathcal{E}$ is unevenly distributed across vehicles, as typically observed in FL setups. Each local dataset $\mathcal{E}_i$ comprises a number of examples E_i of the form $(x_h, y_h), \forall h = 1, \ldots, E_i$, where x_h contains the Lidar point cloud while y_h the corresponding real road actor category. The overall number of available examples in the whole scenario is $E = |\mathcal{E}| = \sum_{i=1}^{N_v} E_i$, with $E_i = |\mathcal{E}_i| \ll E, \forall i \in \mathcal{V}$.

2.2. V2X signaling of ML model parameters

The parameters of the ML model are exchanged through V2V links according to the specific communication standard. Here, we employ the European Telecommunications Standards Institute (ETSI) standard TR 103 562 [44], which rules the communication among road users at application level for cooperative Intelligent Transportation System (ITS) services. In particular, we propose to use the Collective Perception Service (CPS) to share the NN parameters among cooperating vehicles. CPM are broadcast messages exchanged in the V2X network for informing each networked entity of possible detected objects by vehicles and/or by infrastructure i.e., RSU [44]. A CPM is composed by an ITS PDU (Intelligent Transportation Systems Packet Data Unit) header and containers defined in [44]. In particular, the Perceived Object Container (POC) is used here for the purpose of exchanging the ML model parameters during the FL process. POC is an optional container typically used for informing the V2X network about objects or events detected by the other vehicles, or the road infrastructure. As an example, for each detected object reported in the POC, the ID number, the time of measurement, the sensor ID, the object data (e.g., position, speed, acceleration, orientation), confidence and type (e.g., pedestrian, car) can be included. Other containers that might provide additional side information supporting the FL process are listed in the following:

- Management Container (MC): it contains mandatory information about the vehicle, its type and position;
- Station Data Container (SDC): an optional container used for providing specific information about the entity that generated the CPM. This container can include the Originating Vehicle Container (OVC) or the Originating RSU Container (ORC) in case the message is generated either by a vehicle or RSU, respectively.
- Sensor Information Container (SIC): it reports information about the sensors mounted on the vehicle or RSU, such as sensor ID, sensor type and detection area. In case of sensor fusion, the information is encapsulated into a single SIC;

The ITS PDU header, MC and SDC are transmitted using 121 bytes, while the SIC and POC dedicate 35 bytes for each sensor/object in the message, respectively [44, 45]. The generation frequency of the CPM is ruled by the choice of T_{CPM} which ranges from 100 ms to 1000 ms, i.e., from a minimum of 1 message/sec up to 10 messages/sec.

For the purpose of exchanging the ML model parameters in the V2X network, we choose the maximum generation frequency, namely 10 messages/sec. In addition, we consider that the CPM messages always include the maximum number of POCs. This corresponds to a payload of 4480 bytes.

3. Federated learning over V2X networks

Decentralized FL policies let the local model parameters be consensually shared and synchronized across multiple vehicles via V2X networking, possibly without relying on the PS orchestration. In particular, in the proposed distributed implementation vehicles combine local models with neighboring ones by average consensus. Next, they update the combined models using an assigned optimizer running on local data. The FL process generally runs for a number of communication rounds and ends when a consensus is obtained, namely when local models converge to a common representation that satisfies a target loss or accuracy [11].

Considering a deep NN composed of N layers, the goal of FL is to learn a global model $\hat{y}(\mathbf{W}; \mathbf{x})$, where $\mathbf{W} = \mathbf{W}^{(N)}$ encapsulates the parameters of the NN, from the available input data $\mathbf{x}$ and for all N layers. Namely, $\mathbf{W}^{(N)} =$

$\left\{\mathbf{w}_n^{\mathrm{T}}, \mathbf{b}_n\right\}_{n=1}^{N}$ with vectors $\mathbf{w}_n$ and $\mathbf{b}_n$ collecting the NN weights and biases of the layer n. At the last layer N, the predictions are obtained by applying a non-linear function $f(\cdot)$ to the weighted sum over the outputs $\mathbf{h}_{N-1}$ of the second last layer as:

$$\hat{y}(\mathbf{W}^{(N)}; \mathbf{x}) = f_N\left(\mathbf{w}_N^{\mathrm{T}}\mathbf{h}_{N-1} + \mathbf{b}_N\right). \tag{1}$$

Similarly, at a generic layer n the predictions $f_n\left(\mathbf{w}_n^{\mathrm{T}}\mathbf{h}_{n-1} + \mathbf{b}_n\right)$ are recursively computed as a function of the predictions from $n = 1$ up to the previous layer $n-1$, while for $n = 0$ we have $\mathbf{h}_0 = \mathbf{x}$. In FL, the parameters $\mathbf{W}$ can be learned by applying a minimization procedure to any finite-sum objective function $L(\mathbf{W})$:

$$\hat{\mathbf{W}} = \underset{\mathbf{W}}{\operatorname{argmin}}\, L(\mathbf{W}) = \underset{\mathbf{W}}{\operatorname{argmin}}\, \underbrace{\sum_{i=1}^{N_v} \rho_i\, L_i(\mathbf{W})}_{L(\mathbf{W})}, \tag{2}$$

where $\rho_i = E_i / E$ and $L_i(\mathbf{W})$ is the local loss of vehicle i:

$$L_i(\mathbf{W}) = \frac{1}{E_i} \sum_{h=1}^{E_i} \ell(x_h, y_h; \mathbf{W}), \tag{3}$$

and $\ell(x_h, y_h; \mathbf{W})$ is the loss computed over the example (x_h, y_h) when the parameters $\mathbf{W}$ hold.

For the specific case of road actor classification herein considered (see Sec. 4.3), the loss $\ell(x_h, y_h; \mathbf{W})$ is computed as in [43] for C different road actor categories. More specifically, the PointNet model uses two additional regularization losses to allow the input and feature transform networks to learn a representation that is invariant to geometric transformations. This is accomplished by constraining the resulting matrices of the input and feature transform networks to be close to orthogonal. Finally, these two losses are added to the conventional categorical cross-entropy and the total loss is used to compute the gradients and update the weights accordingly.

Algorithm 1 Consensus-driven Federated Averaging

```
procedure C-DFA($\mathcal{V}_{t,i}, \sigma, Q$)
    initialize $\mathbf{W}_{0,i} \leftarrow$ vehicle $i$
    initialize $\mathbf{m}_{0,i} \leftarrow 0$
    initialize $\mathbf{v}_{0,i} \leftarrow 0$
    for each round $t = 1, 2, ...$ do                         ▷ Training loop
        receive$\{\mathbf{W}_{t,j}^{(Q)}\}_{j \in \mathcal{V}_{t,i}}$                         ▷ RX Q layers
        $\psi_{t,i}^{(Q)} \leftarrow \sigma_{i,i}\mathbf{W}_{t,i}^{(Q)}$
        for all vehicles $j \in \mathcal{V}_{t,i}$ do
            $\psi_{t,i}^{(Q)} \leftarrow \psi_{t,i}^{(Q)} + \sigma_{i,j}\mathbf{W}_{t,j}^{(Q)}$
        end for
        $\psi_{t,i} \leftarrow [\mathbf{W}_{t,i}^{(P)}, \psi_{t,i}^{(Q)}]$
        $\mathbf{W}_{t,i}$ = ModelUpdate($\psi_{t,i}$)       ▷ Update all layers
        send$\left(\mathbf{W}_{t,i}^{(Q)}\right)$               ▷ TX Q layers to neighbors
    end for
end procedure
procedure ModelUpdate($\psi_{t,i}$)                           ▷ Local Adam
    $\mathcal{B} \leftarrow$ mini-batches of size B
    for batch $b \in \mathcal{B}$ do
        $\mathbf{m}_{t+1,i} \leftarrow \beta_1\mathbf{m}_{t,i} + (1-\beta_1)\nabla L_{t,i}(\psi_{t,i})$
        $\mathbf{v}_{t+1,i} \leftarrow \beta_2\mathbf{v}_{t,i} + (1-\beta_2)\nabla^2 L_{t,i}(\psi_{t,i})$
        $\psi_{t+1,i} \leftarrow \psi_{t,i} - \frac{\sqrt{1-\beta_2^t}}{1-\beta_1^t} \cdot \frac{\mathbf{m}_{t+1,i}}{\sqrt{\mathbf{v}_{t+1,i}} + \delta}$
    end for
end procedure
```

3.1. Consensus-driven federated learning (C-FL)

We consider the decentralized FL system in Fig. 1, where vehicles learn the global objective (2) by relying only on local computations and mutual exchange of the ML parameters through V2V communications. For vehicle i at time t, the local ML model parameters are represented as $\mathbf{W}_{t,i}$. We propose a modular approach in which the federated learning process targets an optimized subset $\mathbf{W}_{t,i}^{(Q)}$ of $Q \leq N$ model layers, while the remaining $N - Q$ ones are learned using local data only and the chosen optimizer. The proposed approach is tailored for the vehicular setup as it minimizes the communication overhead by limiting the number of ML parameters to be exchanged on each FL round. Without leading in generality, we choose to federate the Q layers closest to the outputs, as more sensitive to unbalanced distributions since they learn features that are specific to the considered dataset [46]. Optimization of the number of layers Q subject to federation is addressed in Sec. 5 for the chosen study. For Q layers, the corresponding model parameters $\mathbf{W}_{t,i}^{(Q)}$ to be exchanged on each communication round are therefore

$$\mathbf{W}_{t,i}^{(Q)} = \left[\mathbf{w}_{N-Q}^{\mathrm{T}}, \mathbf{b}_{N-Q}, \ldots, \mathbf{w}_N^{\mathrm{T}}, \mathbf{b}_N\right], \tag{4}$$

with same definitions for weights $\mathbf{w}_n$ and biases $\mathbf{b}_n$. The remaining layers are learned using local data only and are here collected in $\mathbf{W}_{t,i}^{(P)} = \left\{\mathbf{w}_n^{\mathrm{T}}, \mathbf{b}_n\right\}_{n=1}^{N-Q-1}$, therefore $\mathbf{W}_{t,i} = [\mathbf{W}_{t,i}^{(P)}, \mathbf{W}_{t,i}^{(Q)}]$.

Decentralized ML model parameter sharing and adaptation is implemented by an average consensus approach. The consensus-driven FL (C-FL) process is summarized in the pseudo-code provided in Algorithm 1 and detailed in the following. On every communication round $t = 1, 2, ...$, vehicle i fuses the ML parameters, namely the Q selected layers, received from its neighbors $j \in \mathcal{V}_{t,i}$ via average consensus:

$$\psi_{t,i}^{(Q)} = \sigma_{i,i}\mathbf{W}_{t,i}^{(Q)} + \sum_{j \in \mathcal{V}_{t,i}} \sigma_{i,j}\mathbf{W}_{t,j}^{(Q)}, \tag{5}$$

where $\sigma_{i,j}$ are the mixing weights for the received models which are chosen as [11]:

$$\sigma_{i,j} = \frac{E_j}{\sum_{j \in \mathcal{V}_{t,i}} E_j}, \tag{6}$$

while for $i = j$:

$$\sigma_{i,i} = \frac{E_i}{\sum_{j \in \mathcal{V}_{t,i}} E_j}. \tag{7}$$

Once the consensus process is completed, the vehicle i fuses the shared parameters $\psi_{t,i}^{(Q)}$ with $\mathbf{W}_{t,i}^{(P)}$ into $\psi_{t,i} = \left[\mathbf{W}_{t,i}^{(P)}, \psi_{t,i}^{(Q)}\right]$ and runs the local optimizer, to minimize the local loss $L_i(\psi_{t,i})$ in (3). Considering the Adam optimizer [47] as being the preferred option for PointNet based architectures [43], this last stage is implemented as $\mathbf{W}_{t+1,i} = \psi_{t,i} - \Delta\psi_{t,i}$ with:

$$\begin{cases} \Delta\psi_{t,i} = \mu_t \cdot \dfrac{\sqrt{1-\beta_2^{\,t}}}{1-\beta_1^{\,t}} \cdot \dfrac{\mathbf{m}_{t+1,i}}{\sqrt{\mathbf{v}_{t+1,i}}+\delta} \\ \mathbf{m}_{t+1,i} = \beta_1 \mathbf{m}_{t,i} + (1-\beta_1)\nabla L_{t,i}(\psi_{t,i}) \\ \mathbf{v}_{t+1,i} = \beta_2 \mathbf{v}_{t,i} + (1-\beta_2)\nabla^2 L_{t,i}(\psi_{t,i}) \end{cases}, \tag{8}$$

where $\mathbf{m}_{t+1,i}$ and $\mathbf{v}_{t+1,i}$ are the estimates of the first and second moment of the gradients $\nabla L_{t,i}(\psi_{t,i})$ at round $t+1$. Other parameters $\beta_1, \beta_2 \in [0,1)$, μ_t and δ are defined in [47]. Model adaptation in (8) is computed over a mini-batch $\mathcal{B}$ (of size B) of local training examples. The new parameters $\mathbf{W}_{t+1,i}$ are forwarded to the neighbors of vehicle i and a new round starts. This procedure is iterated until the parameters $\mathbf{W}_{t,i}$ converge to the desired loss values.

3.2. Empowering FL by 6G V2X communications

Consensus driven FL methods rely on an intensive use of low-latency V2X communications for ML model sharing. Communication-efficient FL designs paired with novel 6G radio technologies are thus expected to bring significant benefits. In what follows, we break down and analyze the key features of 6G that are expected to play a pivotal role for massive deployment of FL in vehicular setups.

Mass-market commercialization of connected vehicles for consumers is quite recent, with the standardization of the WiFi-based IEEE 802.11p protocol [48]. As an alternative, the cellular technology of 3GPP Rel. 14 [49], known as LTE-V2X, has also been designed as to accommodate for vehicular applications too. However, moderate availability of spectrum cannot afford the development of enhanced connected mobility services, as planned in 5G V2X [50, 51]. Moreover, the overcrowded bandwidth in the ultra-high and super-high frequency bands (up to 30 GHz) has pushed to investigate millimeter waves (mmWaves) as candidate mean of transmissions. Experimental analyses and proof of concepts validated and confirmed their effectiveness, also in vehicular environments [52–56]. At present, the latest release of 5G New Radio (NR), i.e., 3GPP Rel. 16, is designed to operate up to 52.6 GHz, but research activities for Rel. 17 plan to extend 5G NR support until 71 GHz [57, 58]. Shifting towards this frequency range is mandated to fulfill the demands of automated mobility applications, where high congestion of connected devices is expected to populate the vehicular environment in the next future, with diverse service requirements and communication capabilities [59]. At the same time, hardware enhancements on antenna systems will introduce beam-based communications (possibly adaptive [60]), not only at the network side (i.e., base station) but also on each road user, as higher frequencies (mmWaves) require a compact physical size of the antenna equipment, affording to fit many radiators in a same device [58]. This lead to an increased spectral efficiency, as well as the possibility to handle multiple communications in parallel by spatial selectivity/diversity.

The forthcoming 6G standard is expected to be the first wireless communication technology that faces an unprecedented density of connected devices, with a variety of service requirements and communication modes, requiring specific intelligent functionalities in local edge clouds targeting low latency [61, 62]. Key performance indicators project the 6G technology to a new level of how wireless communications are conceived, paving the way for distributed processing solutions. 6G targets to guarantee a massive data rate for each connection (1 Tb/s) and an end-to-end latency of 0.1 ms [63], turning out to be the first communication technology to truly enable FL over large scale. Consensus-based approaches will largely benefit from almost *zero-latency* 6G V2V communications, letting vehicles to mutually share information on the individual ML models, increasing the availability of data and improving the overall accuracy of the specific algorithm or model.

4. Road actor classification: FL framework, vehicular scenario and dataset

In this section, we present the FL framework tool for road actor classification, the proposed vehicular setting, the Lidar dataset and the PointNet model considered for training and performance validation.

As discussed in Sec. 3, vehicles collectively optimize the ML model for classification, relying only on local optimization and V2X communications for exchanging the model updates. Each vehicle implements a two-stage point cloud processing for classifying the road actors, as depicted in Fig. 2. At first, the Lidar point clouds are processed by a bounding box extractor to obtain object detection and segmentation information (bounding box subsystem). To densify each point cloud, we firstly aggregate 10 Lidar sweeps for each available point cloud and extract the ones that fall within the box. Next, the filtered point clouds are fed to a PointNet-based ML model for obtaining the actual road actors predictions (classification subsystem). Here, FL is applied only to the classification subsystem, while the bounding boxes are processed locally without the cooperation of other vehicles. This choice is shown as practical enough to minimize the communication overhead and limit the ML model size to reasonable complexity.

The overall FL framework, its main components and the related configuration, are detailed in the following sections. First, the vehicular scenario and the virtual network environment are described in Sec. 4.1. Next, extraction and preprocessing of the point cloud data from the nuScenes dataset

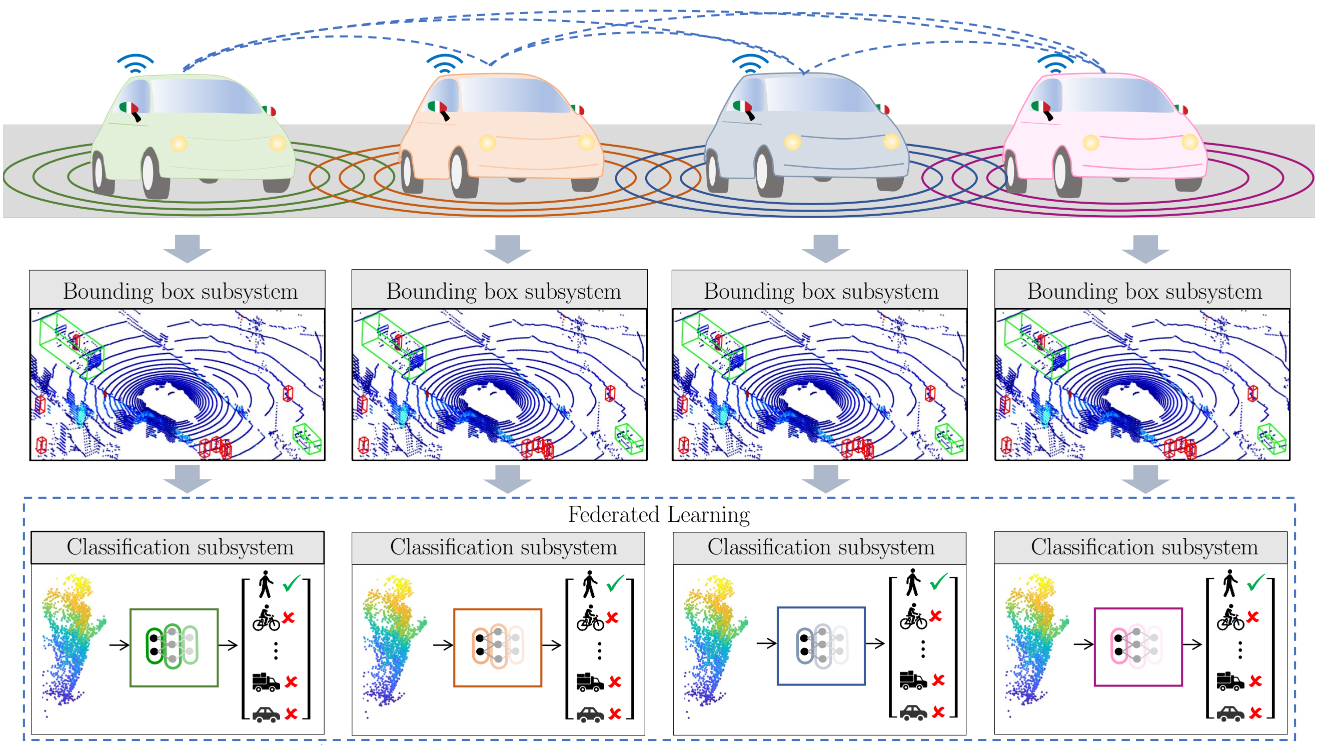


Figure 2: Road actor classification implemented at each vehicle. The classifier consists of 2 subsystems: i) the bounding box segmentation and ii) the classifier based on the PointNet model. FL is applied for distributed training of the PointNet NN model parameters.

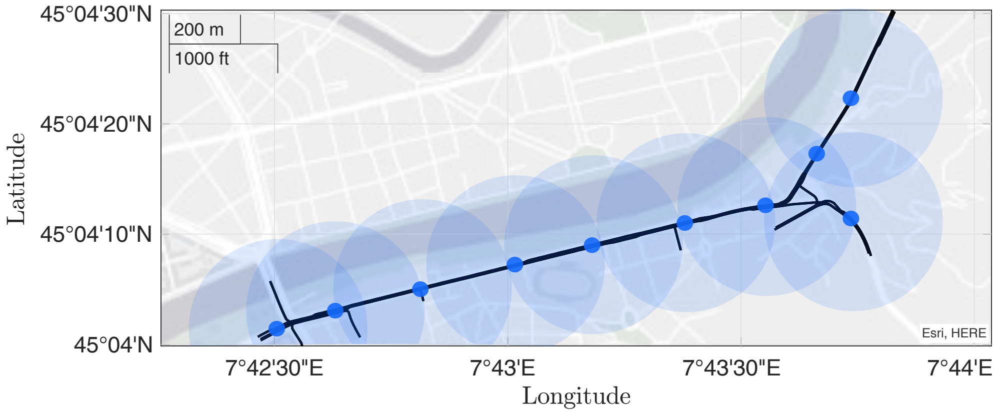


Figure 3: Simulated scenario over an urban arterial in Turin, Italy. Blue dots report vehicles across the road with their respective communication ranges R_c, indicated with the shaded blue circle.

are addressed in Sec. 4.2. Finally, the ML model is defined in Sec. 4.3 and adapted to the considered FL framework.

4.1. Simulation of the FL vehicular scenario

The FL framework is validated using vehicular traffic data simulated over the urban arterial of Turin, Italy, as reported in Fig. 3. Vehicles' trajectories are simulated using the Advanced Interactive Microscopic Simulator for Urban and Non-Urban Networks (AIMSUN) onto this map [64]. This provides realistic traffic conditions and allows us to evaluate the influence of the vehicles' mobility onto the FL training procedure.

On top of AIMSUN framework, we developed a virtual environment that allows to deploy vehicles acting as mobile devices that learn over a configurable subset of the data. To simulate the FL process, the Lidar datasets, described in Sec. 4.2, are distributed to the virtual mobile devices according to an arbitrary policy (both IID and non-IID are supported). Before feeding the training examples to the local NN PoinNet model, described in Sec. 4.3, we normalize the input point clouds such that they are contained into a unit area sphere. In addition, as the NN model requires a fixed number of points for the input point cloud, we upsample/downsample x_h to contain 2048 points.

The NN parameters exchange in FL is performed through the usage of CPM messages in the V2X network, as previously discussed in Sec. 2.2. Once the local optimization step is performed, vehicles share the selected Q layers of the ML model by encapsulating the parameters into the POCs of the CPM. Considering the FL process, and that the maximum number of objects that can be reported in the POC is 128, the minimal set of information included in the container are:

- the subset, or fragment, of the local ML parameters from the Q selected layers (see Sec. 3);
- the corresponding identification (ID) of the fragment and hyperparameters (layer type, weights and biases);
- the time reference: learning epoch or FL round;
- the local loss $\mathcal{E}$ used as model quality indicator.

Each message has an overall payload of 4480 bytes and it is transmitted every 0.1 s. To exchange the selected layers, a number of CPM messages (fragments) should be multiplexed, depending on the model size and Q. In what follows we also assume that CPM messages are always correctly received. Nevertheless, the proposed framework can be generalized to account for packet drops.

The V2X connectivity graph is extracted using the actual positions of the vehicles and their communication range R_c. In the following section, we characterize the performances of FL considering different communication ranges. This allows us to characterize how the FL process responds to different connectivity graphs and how the performances are affected by the number of neighbors observed by each vehicle.

4.2. Vehicular Lidar dataset

FL has been simulated using the Lidar point cloud dataset of nuScenes [65]. nuScenes is a large-scale autonomous driving dataset published by nuTonomy in 2019, consisting of 1000, 20 second-long, scenes of driving captured across Boston and Singapore. The dataset offers scenes with different weather conditions, traffic densities as well as lighting. The vehicle used for the recordings is equipped with a full sensors' suite, composed by a 360 degree Lidar, 5 long-range Radars, 6 cameras and an Inertial Measurement Unit (IMU) sensor. All objects detected in the scenes are annotated manually at a frequency of 2 Hz to ensure high accuracy. Each annotated object consists of a 3D bounding box as well as the object's category. In what follows, the overall database used for classification is composed by 9000 examples, equally divided for each one of the 6 considered categories: pedestrian, car, bus, bicycle, barrier and traffic cones[1]. The h-th training example contains the point cloud x_h that fall into the box and the corresponding category y_h.

For evaluating the FL procedures, we use a separated validation dataset composed by 2400 examples, equally divided into the 6 road actor classes. The same pre-processing procedure used for the training dataset, i.e. upsample/downsample and normalization stages, is also applied to the validation database to assess the performances under similar conditions.

Table 1
PointNet model parameters.

Layer	Filters/Neurons	Parameters	Federated Layers
CN	8	32	$Q = 20$
CN	16	144	
CN	128	2176	
FC	64	8256	
FC	32	2080	$Q = 16$
FC	9	297	
CN	8	32	
CN	8	72	
CN	8	72	$Q = 12$
CN	16	144	
CN	128	2176	
FC	64	8256	
FC	32	2080	$Q = 8$
CN	8	72	
CN	16	144	
CN	128	2176	$Q = 4$
FC	64	8256	
FC	32	2080	
FC	6	198	

4.3. ML model and FL adaptation

The PointNet ML model [43], depicted in Fig. 4, is used for 3D shape classification and segmentation from point cloud data. The classification network takes as input the point clouds, applies an input and feature transformation, and then aggregates the learned features by a max pooling operation. Classification scores are obtained by a Multi-Layer Perceptron (MLP) followed by a softmax operation. As this model was originally developed for the ModelNet40 dataset, which takes into account 40 different classes, we adapted the model structure and parameters to reflect the considered learning tasks. In particular, the number of filters used in the Convolutional (CN) layers and the number of neurons in the Fully Connected (FC) layers are reduced by a factor of 8, the dropout layers have been removed and the final layer is composed of $C = 6$ output neurons. Furthermore, the Batch Normalization (BN) layers apply a momentum factor of 0.9. This means that, during inference, more importance is given to the average statistics computed over all batches rather than the instantaneous ones for the current batch.

In the proposed implementation, FL procedures are not applied to the BN layers, so the statistics are only updated locally without the cooperation of other vehicles. With such modifications, the number of trainable layers is $N = 20$ (9 FC and 11 CN layers) while the total number of model parameters that can be exchanged is 40855. Table 1 summarizes the parameters for the modified PointNet architecture and details the various choice of Q for the modular C-FL approach. Note that FC layers include bias parameters while CN do not.

5. Numerical results

In this section, we present the numerical results for the evaluation of the proposed C-FL framework. The approach

[1] Note that 23 road actors categories are available, ranging from large vehicles (trailer and truck) to little objects (traffic cones).

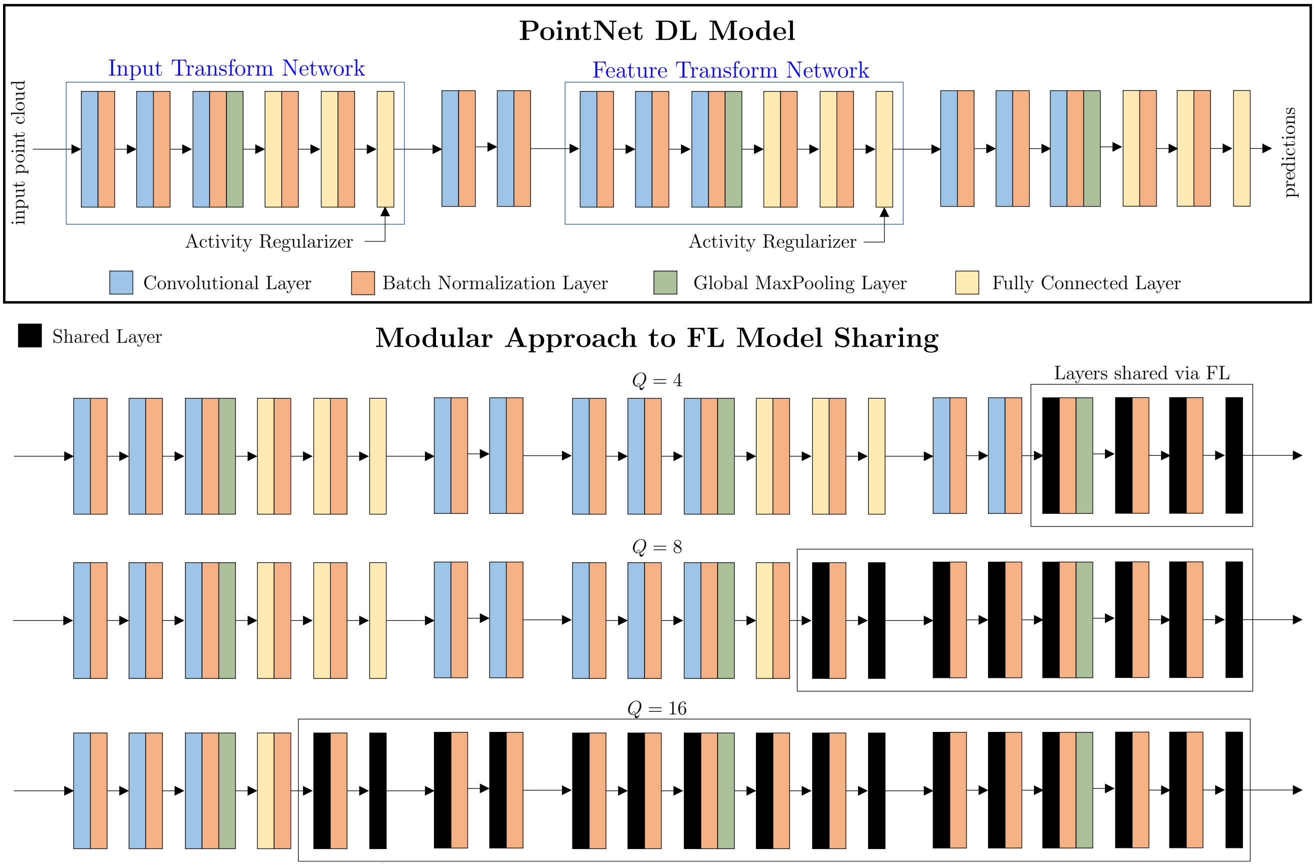


Figure 4: PointNet ML model and FL model sharing via modular approach ($Q = 4, 8, 16$).

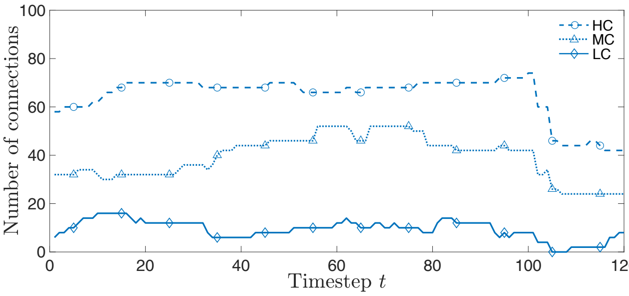


Figure 5: Temporal evolution of the number of aggregated V2V connections.

is validated in terms of loss, accuracy and convergence time according to the V2X network setup previously described. We also consider the performance comparison with ego learning, featuring vehicles that learn from local training data only disabling cooperation, PS based FL [13] as well as conventional DML with centralized learning and fusion at the data center.

In the following, Sec. 5.1 details the setting and system parameters used for the simulations. Sec. 5.2 describes the results for the IID case, Sec. 5.3 for the non-IID one, while Sec. 5.4 illustrates a 6G-enabled FL case. Sec. 5.5 presents the validation of a continual learning case where the training process is periodically updated using new input data collected by the vehicles. This latter scenario is compared to conventional one-time learning. Finally Sec. 5.6 discusses relevant implementation issues and resource constraints.

5.1. System parameters

We use a vehicular network of $N_v = 10$ vehicles for assessing the modular C-FL method and characterize the impact of the V2X end-to-end communication latency over the training procedure. The V2X connectivity graph at time t depends on the communication range R_c and vehicle positions $\mathbf{p}_{t,i}, \forall i \in \mathcal{V}$, extracted using the AIMSUN simulator. The entry (i, j) of the V2X connectivity matrix is equal to 1 if vehicles i and j satisfy the constraint $\|\mathbf{p}_{t,i} - \mathbf{p}_{t,j}\| \leq R_c$, $\forall i, j \in \mathcal{V}$ otherwise it is equal to 0. We consider three communication range values $R_c = \{100, 500, 1000\}$ m for evaluating the impact of the connectivity on the C-FL procedure, referred to as low (LC), medium (MC) and high connectivity (HC) cases, respectively. Fig. 5 reports the total number of active V2V connections for the three connectivity cases (LC, MC, HC) and for each timestep t. As expected, in the LC case vehicles interact with few neighbors. Moreover, for some timesteps, i.e., from $105 \leq t \leq 108$, they do not implement the model exchange phase as no connections exist. On the other hand, medium (MC) and high (HC) connectivity always guarantee cooperation among vehicles.

The overall dataset presented in Sec. 4 is distributed to the vehicles according to some predefined policies before the training starts. Here, we consider two partitioning strategies with homogeneous (IID) and non-homogeneous (non-IID) statistical distribution of data over vehicles. More specifically, when the dataset is distributed according to an IID policy, vehicles hold the same number of examples for each of the $C = 6$ available classes, while in the non-IID they do not retain examples for some classes. Local optimization of the ML model is performed using a batch size of $B = 30$ examples and the Adam optimizer of (8) is configured with $\mu_t = 10^{-4}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\delta = 10^{-7}$.

The C-FL algorithm is implemented by applying the federated optimization to a variable number $Q \leq N$ of layers in the NN, starting from the outer ones, i.e., $N = N, N - 1, \dots, N - Q + 1$. More specifically, performances are analyzed by varying the fraction $M = Q/N$ of the layers subject to the C-FL process. This ranges from $M = 20\%$, (i.e., corresponding to $Q = 4$ outer layers) up to $M = 100\%$ (i.e., all layers, apart the BN ones). C-FL is compared against a centralized (vanilla) FL tool that relies on a PS to update the received models from vehicles. Notice that, to ease the comparison, the centralized FL setup also applies the same modular technique as defined in C-FL (Sec. 3) for exchanging configurable fragments of the ML parameters. The PS server is assumed to be always connected with all available vehicles for every communication round. The performance of opportunistic learning, referred to as Ego Learning (EL), is also considered as benchmark. In this case, vehicles do not cooperate and use only their local dataset $\mathcal{E}_i$ for optimizing the PointNet ML model for classification. Finally, C-FL is compared with a conventional DML technique, referred to as Centralized Learning (CL), where a central orchestrator, namely a data center, collects all the data produced by the vehicles and implements the model optimization without any help from the vehicles.

For all the above defined cases, we evaluate the time required for completing the training procedure by considering the specific operations that are performed on every epoch. In FL and C-FL, the total time required for completing a communication round can be broken down into the time for running the local optimizer, the time for exchanging the model parameters, either with PS (centralized) or mutually (C-FL) and the time for applying the aggregation policy, or consensus average, respectively. For CL, a preliminary phase is executed for uploading the raw data, i.e., the Lidar point clouds, on the data center at the beginning of the training procedure. Finally, in EL the vehicles optimize their own NN using only their local data. Therefore, the time required for completing one epoch corresponds to the time used by the local optimizer.

Table 2 reports the number of parameters exchanged at every communication round, the corresponding number of CPM messages and V2X end-to-end time required to implement C-FL and FL. For CL we quantified the time required for the initial raw data upload at the data center. We assume that the Lidar point clouds are transmitted using an encoding of 4 bytes/parameter while the ML parameters need 8 bytes/parameter. Notice that CL also requires to fed back the trained model to all participating vehicles. For what concerns the local computation times, the time needed for completing a local Adam stage is 0.2 s for all FL designs and EL. For CL, this is one order of magnitude higher, i.e., 2 s, as data fusion is considered for all $N_v = 10$ vehicles. Finally, the consensus-driven aggregation phase of C-FL depends on Q and the number of neighbors, namely the number of models received. Computations range from a minimum of 6 ms, when $Q = 4$ is used with 1 neighbor, up to 166 ms for $Q = 20$ and 10 neighbors.

Table 2
V2X communication resources and times.

Approach	Parameters	Payload	Messages	V2X time
	[#]	[kB]	[#]	[s]
C-FL $M = 20\%$	12710	101.68	23	2.3
C-FL $M = 40\%$	17118	136.94	31	3.1
C-FL $M = 60\%$	27766	222.12	50	5
C-FL $M = 80\%$	30247	241.98	55	5.5
C-FL $M = 100\%$	40855	326.84	73	7.3
CL (IID)	1658880	6635.5	1482	148.2
CL (non-IID)	1228800	4915.2	1098	109.8

5.2. Statistically homogeneous (IID) scenario

First, we analyze the performance of C-FL when the data stored at each vehicle are IID-distributed. In particular, all vehicles hold a local dataset comprised of $\rho_k = 3\%, \forall i \in \mathcal{V}$, equally divided for each one of the available classes. Validation loss and accuracy are computed by using the validation dataset presented in Sec. 4.

Fig. 6 reports the validation loss for C-FL with $M = 100\%$ (Fig. 6a), $M = 80\%$ (Fig. 6b), $M = 60\%$ (Fig. 6c), $M = 40\%$ (Fig. 6d), and $M = 20\%$ (Fig. 6e), while Fig. 7 presents the corresponding validation accuracy for C-FL with $M = 100\%$ (Fig. 7a), $M = 80\%$ (Fig. 7b), $M = 60\%$ (Fig. 7c), $M = 40\%$ (Fig. 7d), and $M = 20\%$ (Fig. 7e). We consider the low (C-FL LC), medium (C-FL MC) and high (C-FL HC) connectivity scenarios as well as the centralized implementation (FL) for comparison. Analyzing the results, it can be seen that EL converges very fast when compared to all other methods. However, the accuracy reached at convergence is low compared to FL and CL, indicating that EL may not be appropriate when high accuracy is needed. CL needs a considerable initial time at the data center for fusing the raw data from the vehicles. On the other hand, it is able to provide highly accurate results outperforming EL. As the data stored at each vehicle increases, CL may be prohibitively limited by such initial communication and fusion stage introducing large delays in the training process. C-FL and centralized FL tools are able to approach the performances of CL when $M = 100\%$, i.e., all layers are exchanged, however they require more time for convergence, namely more communication and learning rounds. This is expected as vehicles rely only on their local database for model optimization holding a much lower pool of examples compared to CL, which has access to the combined database of all vehicles. On the other hand, raw data transmission might be sometimes unfeasible also violating privacy. Being not limited by the size of the data, nor requiring the same data to be disclosed to third parties, FL policies can be thus considered as promising for big-data vehicular applications.

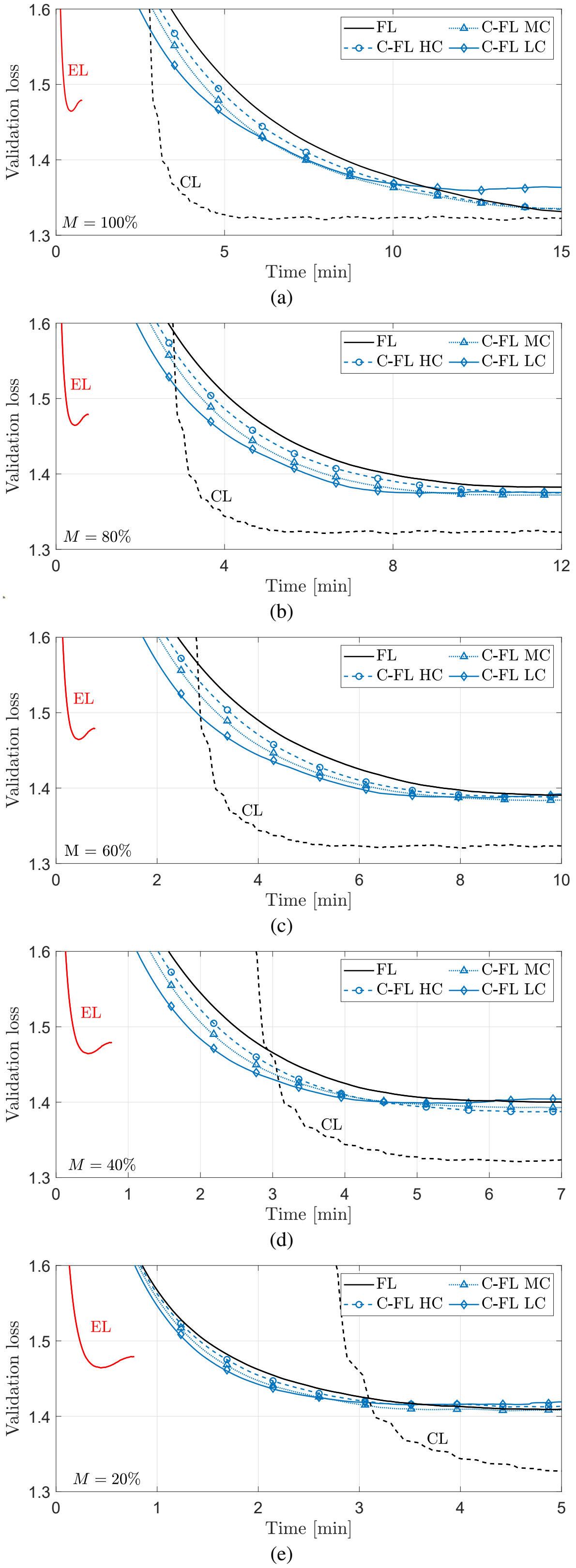
Focusing now on the performances of C-FL for varying levels of connectivity, it can be seen that the number of connections heavily influences the convergence speed and classification accuracy. For $M = 100\%$, centralized FL presents the lowest convergence rate, followed by C-FL HC, C-FL MC and C-FL LC. Similar results are obtained also for M ranging from 20% up to 80%. However, in most of the cases, C-FL LC case has higher validation loss (or lower accuracy) compared with other setups, suggesting that a high enough level of cooperation among vehicles is needed to improve the classifier performance. Results also indicate that, beside decreasing the overhead for each communication stage, it is also important to provide more informative updates at each round [12]. Accurate local model optimization should be preferred initially to let the vehicles reach a rough local solution before starting the consensus aggregation policy. Once vehicles have exploited their local data enough, consensus can be enabled to improve the accuracy.

Looking at the performances for varying number Q of layers subject to federation, results detail that the best accuracy is achieved when all layers $M = 100\%$ are shared among vehicles, as expected. However, communication efficient designs, i.e., $M < 100\%$, might improve convergence time in exchange for accuracy penalties. For example, the performances for $M = 60\%$ and $M = 40\%$ are similar, thereby $M = 40\%$ should be preferred to maximize efficiency. $M = 20\%$ can be chosen at the cost of 3% lower accuracy compared to $M = 40\%$. On the other hand, choosing $M = 80\%$ rather than $M = 100\%$ induces an accuracy penalty of 5%, however less V2X resources are used, i.e., the payload required for transmitting model updates reduces by almost 30%.

5.3. Statistically heterogeneous (non-IID) scenario

FL setups are typically characterized by non-IID information across participating devices in the training process. Since data is collected independently on each vehicle, statistical heterogeneity may arise as a result of the vehicle-specific data collection process. Moreover, malicious data injection and faulty nodes may affect the performances of FL systems, especially in non-IID settings. To alleviate such detrimental effects, authentication procedures or blockchain technologies can be leveraged [66, 67]. This section analyzes the performance of the C-FL approach under non-IID data partitioning across vehicles. In particular, all vehicles hold a local dataset comprised of $\rho_i = 2.5\%, \forall i \in \mathcal{V}$, equally divided among 5 of the 6 classes available. As done before, performances are assessed by computing the validation loss, accuracy and convergence time, averaged over all vehicles.

Fig. 8 reports the validation loss for C-FL with $M = 100\%$ (Fig. 8a), $M = 80\%$ (Fig. 8b), $M = 60\%$ (Fig. 8c), $M = 40\%$ (Fig. 8d), and $M = 20\%$ (Fig. 8e). Fig. 9 presents the validation accuracy for C-FL with $M = 100\%$ (Fig. 9a), $M = 80\%$ (Fig. 9b), $M = 60\%$ (Fig. 9c), $M = 40\%$ (Fig. 9d), and $M = 20\%$ (Fig. 9e). As presented in the previ-



(a)

(b)

(c)

(d)

(e)

Figure 6: Analysis of the validation loss under IID data partitioning for varying number of federated layers and connectivity: (a) $M = 100\%$, (b) $M = 80\%$, (c) $M = 60\%$, (d) $M = 40\%$, (e) $M = 20\%$.

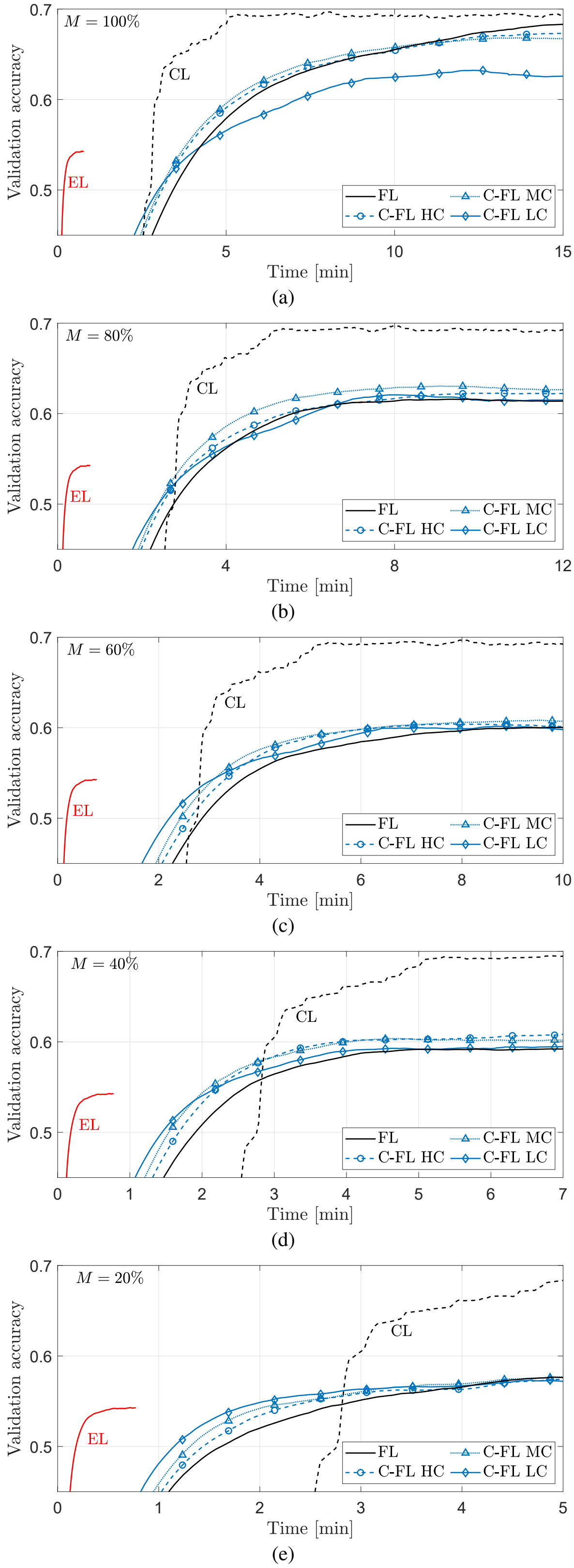


Figure 7: Analysis of the validation accuracy under IID data partitioning for varying number of federated layers and connectivity: (a) $M = 100\%$, (b) $M = 80\%$, (c) $M = 60\%$, (d) $M = 40\%$, (e) $M = 20\%$.

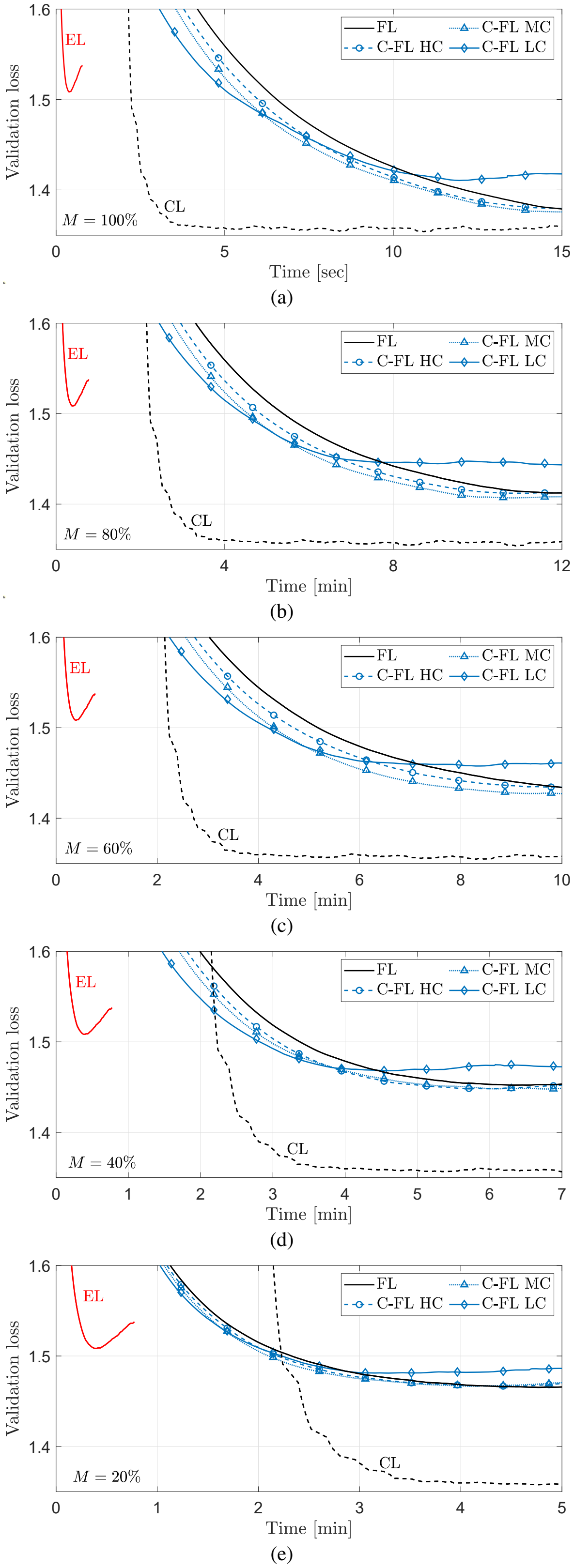


Figure 8: Analysis of the validation loss under non-IID data partitioning for varying number of federated layers and connectivity: (a) $M = 100\%$, (b) $M = 80\%$, (c) $M = 60\%$, (d) $M = 40\%$, (e) $M = 20\%$.

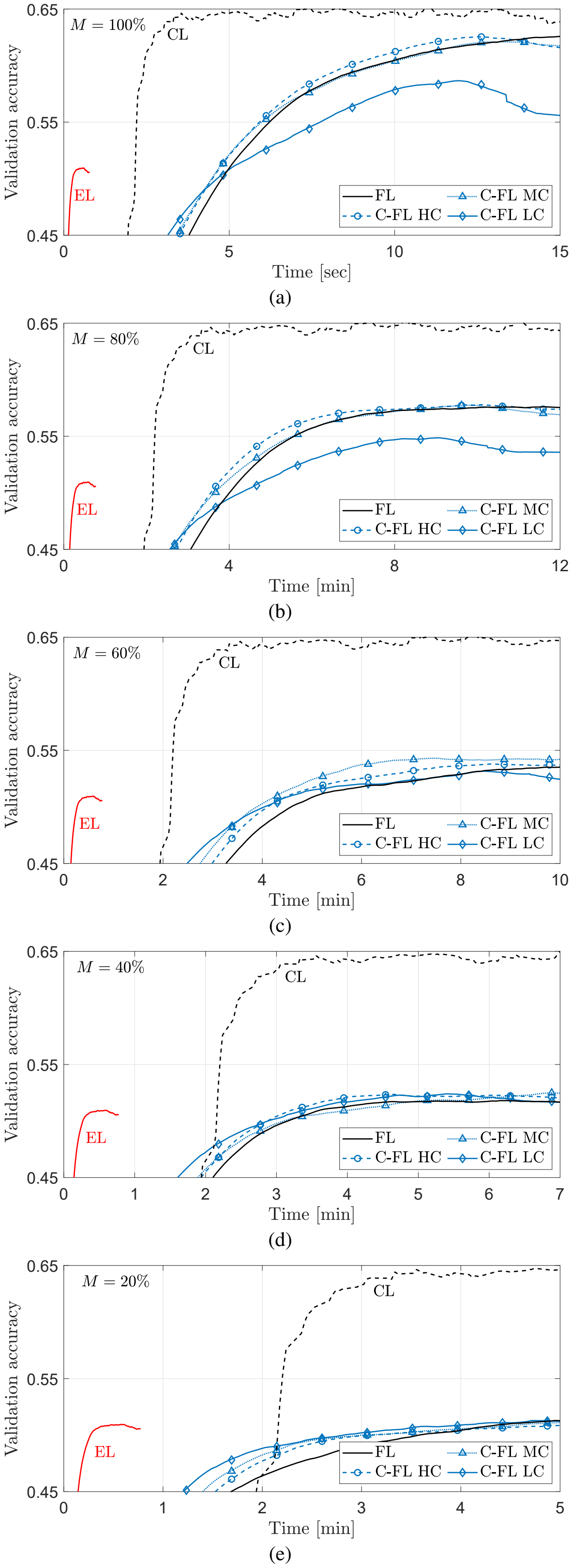


Figure 9: Analysis of the validation accuracy under non-IID data partitioning for varying number of federated layers and connectivity: (a) $M = 100\%$, (b) $M = 80\%$, (c) $M = 60\%$, (d) $M = 40\%$, (e) $M = 20\%$.

Table 3
Validation loss times.

Approach		**Target val. loss times [min]**			
		1.55	**1.50**	**1.45**	**1.40**
EL		0.21	N.A.	N.A	N.A
C-FL LC	$M = 20\%$	1.40	2.20	N.A	N.A
	$M = 40\%$	1.99	2.80	N.A	N.A
	$M = 60\%$	2.98	4.15	N.A.	N.A.
	$M = 80\%$	3.25	4.43	6.79	N.A.
	$M = 100\%$	3.99	5.40	7.84	N.A.
C-FL MC	$M = 20\%$	1.45	2.09	N.A.	N.A.
	$M = 40\%$	2.23	2.93	5.75	N.A
	$M = 60\%$	3.27	4.27	6.27	N.A
	$M = 80\%$	3.46	4.54	6.33	N.A
	$M = 100\%$	4.40	5.69	7.50	11.00
C-FL HC	$M = 20\%$	1.46	2.19	N.A.	N.A.
	$M = 40\%$	2.30	3.07	5.36	N.A.
	$M = 60\%$	3.48	4.67	7.05	N.A.
	$M = 80\%$	3.77	4.86	6.75	N.A.
	$M = 100\%$	4.69	5.99	7.81	11.19
FL	$M = 20\%$	1.52	2.34	N.A.	N.A.
	$M = 40\%$	2.55	3.38	N.A.	N.A.
	$M = 60\%$	3.88	5.17	8.02	N.A.
	$M = 80\%$	4.19	5.38	7.47	N.A.
	$M = 100\%$	5.25	6.55	8.52	12.05
CL		2.18	2.24	2.43	2.71

ous section, the FL tools are evaluated considering low (C-FL, LC), medium (C-FL, MC) and highly dense (C-FL, HC) connectivity cases. These are compared with EL and CL using the same non-IID data partitions. Compared to the IID case, EL shows clear signs of overfitting, indicating that relying only on local data for optimizing the ML model may not be appropriate and the learned models do not generalize well to the considered classification task. It is worth noting that, with respect to the IID data partitioning policy, here the classification accuracy obtained by all methods is lower.

Convergence time and accuracy are affected by connectivity scenarios LC, MC and HC. Similarly as before, C-FL LC gives the lowest convergence time while FL the highest. C-FL MC or HC setups should be preferred to LC when data is non-IID: cooperation among vehicles is thus more critical when data is unbalanced across vehicles.

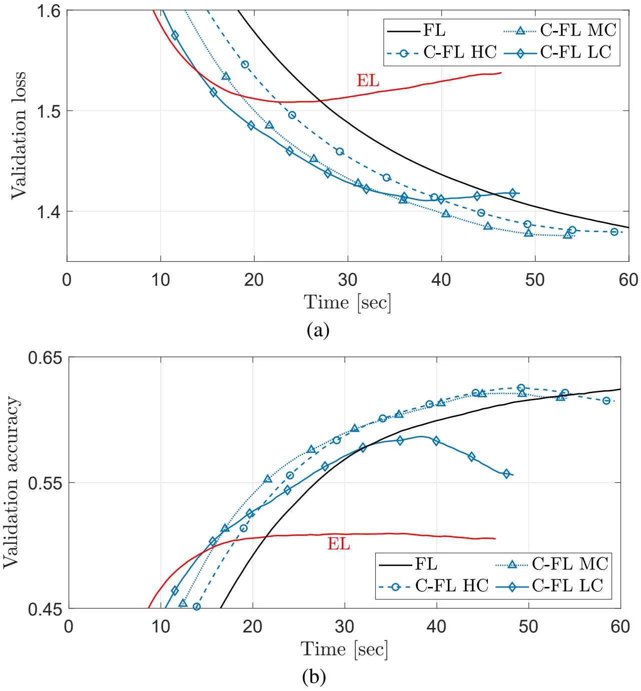
Implementing federation over all the model layers ($M = 100\%$) gives again the highest/lowest validation accuracy/loss. However, as opposed to the IID case, the performances always improve with M. Indeed, in this setting, $M = 60\%$ should be preferred with respect to $M = 40\%$ to augment the classification accuracy when the communication resources of the V2X network permit this choice. In Table 3 and Table 4 we report the time required for all methods to reach a target validation loss and accuracy, respectively. These tables are chosen to highlight the performance comparison between C-FL, FL, CL and ego approaches.

5.4. FL in 6G V2X network

To conclude the analysis for non-IID data, we now characterize the FL policies and EL, considering a realistic end-

Table 4
Validation accuracy times.

Approach		Target val. acc. times [min]			
		0.45	**0.50**	**0.55**	**0.60**
EL		0.14	0.50	N.A.	N.A.
C-FL LC	M = 20%	1.22	2.84	N.A.	N.A.
	M = 40%	1.58	2.86	N.A.	N.A.
	M = 60%	2.44	4.06	N.A.	N.A.
	M = 80%	2.56	4.24	N.A.	N.A.
	M = 100%	3.09	4.63	7.72	N.A.
C-FL MC	M = 20%	1.41	2.99	N.A.	N.A.
	M = 40%	1.87	3.11	N.A	N.A.
	M = 60%	2.27	3.91	N.A	N.A.
	M = 80%	2.67	3.66	5.54	N.A.
	M = 100%	3.36	4.53	5.45	9.45
C-FL HC	M = 20%	1.51	3.10	N.A.	N.A.
	M = 40%	1.95	2.89	N.A.	N.A.
	M = 60%	3.02	4.12	N.A.	N.A.
	M = 80%	2.68	3.57	5.06	N.A.
	M = 100%	3.52	4.43	5.86	8.59
FL	M = 20%	1.52	2.34	N.A.	N.A.
	M = 40%	2.55	3.38	N.A.	N.A.
	M = 60%	3.88	5.17	N.A.	N.A.
	M = 80%	4.19	5.38	7.47	N.A.
	M = 100%	5.24	6.55	8.52	12.05
CL		1.93	2.14	2.21	2.43



(a)

(b)

Figure 10: Performance analysis for C-FL with $M = 100\%$ over 6G V2X networks: (a) validation loss, (b) validation accuracy.

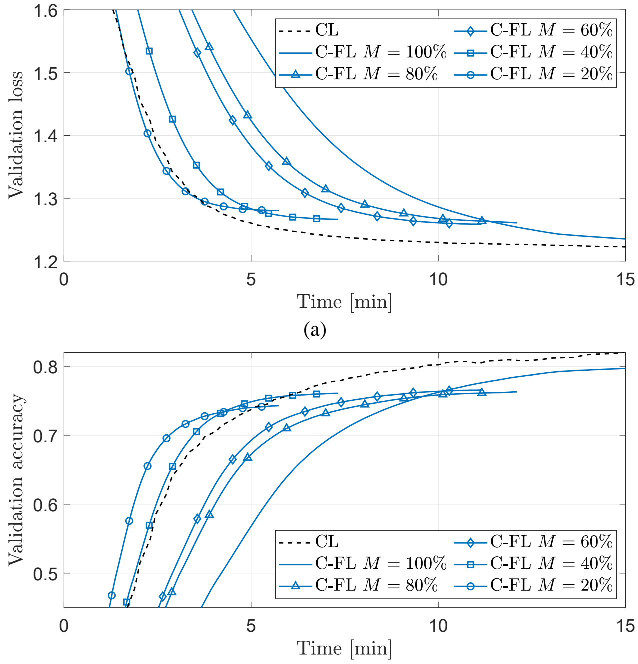
to-end latency to better adhere to the envisioned communication capabilities of 6G V2X networks. In the previous examples, the model exchange and/or raw data upload rely on standardized CPM messages that cannot meet the challenging requirements foreseen for fully autonomous driving scenarios. We now resort to an enhanced 6G V2X setup where the model parameters are exchanged using a single message within 1 ms [63] thanks to the ultra wide band V2X connectivity. The payload required for exchanging all the model parameters for C-FL is 326.84 kB while the data rate is 2.61 Gbps, i.e., the payload in bit divided by 1 ms. Such values are compliant with the envisioned 6G V2X performance which target 1 Tbps [61]. The comparison focuses on C-FL with $M = 100\%$ and EL, by evaluating the convergence time as well as the validation loss and accuracy.

Fig. 10a reports the validation loss while Fig. 10b the validation accuracy for C-FL under low (C-FL LC), medium (C-FL MC) and high connectivity (C-FL HC) settings. These are again compared with centralized FL and EL. Using V2X communications now rooted in the 6G paradigm, the convergence time is 15 times lower compared to the previous cases, this corresponds to a learning time scaling down from roughly 15 minutes to 1 minute. C-FL, paired with novel 6G, is thus able to outperform EL by a much larger margin while keeping the overall time required for completing the FL process comparable to EL.

5.5. Continual Learning

Real world vehicular applications are characterized by a continual and online sensor data acquisition that changes in response to the specific environment in which vehicles are deployed. As the time passes, data is continually gathered while periodic re-training of the ML models should be carried out to adapt to the changing environment and exploit this newly acquired data to further refine the models. To better address this aspect, here we evaluate the performances of C-FL and CL in a continual learning setting where vehicles periodically collect new Lidar data samples. At the start of the training phase, the vehicles have access to 30 randomly drawn examples chosen from an IID or a non-IID dataset. The local IID data available at the vehicles is composed by $\rho_i = 5\%, \forall i \in \mathcal{V}$, equally divided among the 6 classes, while the non-IID one comprises $\rho_i = 4.1\%, \forall i \in \mathcal{V}$, evenly partitioned across only 5 classes. The continuous data gathering process is here simulated using the following procedure. Every 5 epochs, corresponding to 23 s, the vehicles randomly select 15 new training examples and add them to the overall training database that is used for local model optimization. This procedure is iterated until all available examples are used. The V2X connectivity chosen is the medium one with $R_c = 500$ m.

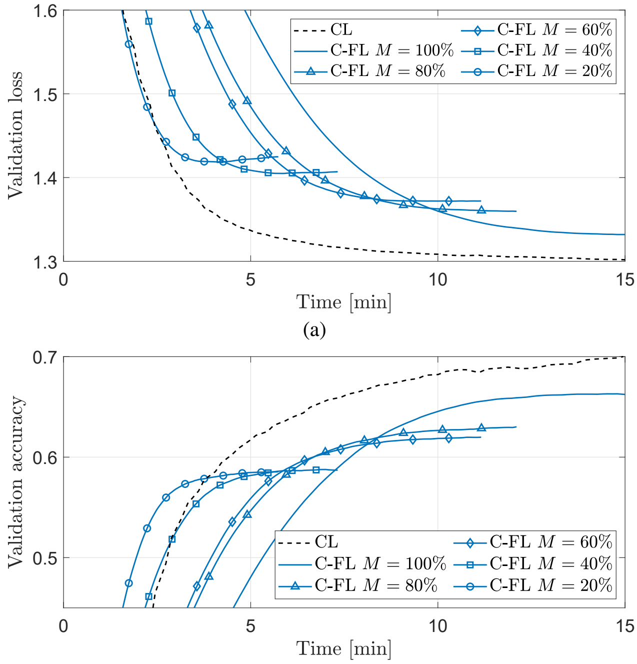
Fig. 11 reports the validation loss (Fig. 11a) and accuracy (Fig. 11b) for the IID case with C-FL ranging from $M = 20\%$ up to $M = 100\%$ and CL, while Fig. 12 shows the validation loss (Fig. 12a) and accuracy (Fig. 12b) for the non-IID one. Analyzing the results, C-FL now provides comparable performances to CL when $M = 100\%$ under IID. On the other hand, when data is non-IID, there exists an accuracy gap between the two. Continual CL now requires the participating vehicles to send updated training examples every time new Lidar point clouds are gathered. The need to implement frequent uplink transmissions introduces further



(a)

(b)

Figure 11: Performance analysis for continual learning with IID data partitioning for C-FL ranging from $M = 20\%$ up to $M = 100\%$ and CL: (a) validation loss, (b) validation accuracy.



(a)

(b)

Figure 12: Performance analysis for continual learning with non-IID data partitioning for C-FL ranging from $M = 20\%$ up to $M = 100\%$ and CL: (a) validation loss, (b) validation accuracy.

delays in the CL process. As a consequence, the convergence times of C-FL and CL are now comparable. Notice that, as more data is produced by the vehicles, CL might be penalized by higher communication overheads than FL. The ML model parameters have also smaller size compared to raw (uncompressed) Lidar point clouds, while they do not need anonymization before transmission.

5.6. FL implementation and resource constraints

Real-world FL setups need to consider several implementation issues, summarized in the following. Maximizing communication efficiency while decreasing convergence time is of paramount importance not only for reducing the network resources, as previously highlighted, but also to minimize the energy footprints of the vehicles participating in the FL process. With this respect, vanilla and consensus-based FL tools must generally limit the V2X channel use only for sharing truly informative updates. For example, during C-FL initialization, a sufficient number of local optimization rounds needs to be implemented on each vehicle so that the local model is accurate enough to initiate the consensus process. Similarly, the FL process should be terminated promptly when no further improvements are experienced. Distributed implementations may also require dedicated set-up phases to distribute the initial model hyperparameters. i.e., DL model structure, while centralized ones can exploit the PS to control the overall initialization stage.

As far as the vehicles' resources are concerned, all FL policies let the vehicles collect their own training data without requiring to store them in a central location. This allows to limit the storage required at the end nodes compared to DML systems. On the other hand, FL requires the vehicles to participate in the model training process. The most computationally intensive operation for FL setups is thus the local model optimization: the vehicles computing hardware should typically support gradient-based optimization, via dedicated processing units or Tensor Processing Units (TPU) [68]. Besides local optimization, the proposed C-FL policy requires each vehicle to reserve additional memory space for storing the model parameters obtained from the neighboring vehicles, and to implement averaging or fusion of the same parameters, i.e., the selected model layers. On the other hand, weighted sum over the received models requires far less computational resources compared to local model optimization and scales linearly as the number of cooperating vehicles increases.

6. Conclusions

In this paper, we explored the potentials of the consensus-driven Federated Learning (C-FL) paradigm in V2X networks to provide communication-efficient distributed training services. In particular, we developed a modular decentralized FL approach for road actor classification, where model sharing can be implemented on a variable number of layers, depending on the required efficiency and bandwidth requirements. The federation process is applied to a PointNet compliant machine learning architecture for classification of road objects using Lidar point clouds as inputs. Model exchange among interconnected vehicles during the FL process relies on a realistic V2X network and uses the standardized CPS service for encapsulating the NN parameters into the CPM messages.

The characterization of the proposed consensus-driven FL (C-FL) approach has been performed by evaluating the accuracy and loss performances as well as the convergence time considering different connectivity and vehicles mobility scenarios, IID and non-IID data partitioning policies, as well as continual learning setups. Experimental results show that the proposed C-FL method is able to outperform ego (EL) approaches by a large margin, while providing comparable performances to centralized (CL) implementations. Results also suggest that consensus among vehicles should be adapted to privilege informative updates or vehicles possessing high quality local models. Furthermore, connectivity among vehicles heavily influences the accuracy and the learning time: C-FL typically performs well in dense networks characterized by a large population of interconnected vehicles.

The analysis has also shown that currently-standardized CPM messages are not appropriate for latency-sensitive vehicular applications, as they might lead to high convergence time or insufficient accuracy in some cases. On the other hand, the proposed FL process truly supports low-latency distributed intelligence when seamlessly integrated with 6G enhanced V2X communications: this is shown as useful to scale down the learning time and meet the challenging requirements foreseen for full self-driving scenarios, often requiring continual learning on large and time-varying data structures.

Further research activities are needed on novel FL and 6G convergent designs. These will possibly embrace new technology enablers, such as multi-connectivity techniques to support efficient long and short range V2V communications. Besides, the implementation of ad-hoc network slicing procedures optimized to sustain the FL process in high mobility scenarios is also promising and could bring further improvements.

Acknowledgment

The authors would like to thank Prof. Francesco Deflorio of DIATI Transport Systems, Politecnico di Torino, for the cooperation in the simulation of the AIMSUN-based vehicular traffic scenario.

References

[1] H. Viswanathan, P. E. Mogensen, Communications in the 6G era, IEEE Access 8 (2020) 57063–57074. doi:10.1109/ACCESS.2020.2981745.

[2] A. Eskandarian, C. Wu, C. Sun, Research advances and challenges of autonomous and connected ground vehicles, IEEE Transactions on Intelligent Transportation Systems 22 (2) (2021) 683–711. doi:10.1109/TITS.2019.2958352.

[3] S. Grigorescu, B. Trasnea, T. Cocias, G. Macesanu, A survey of deep learning techniques for autonomous driving, Journal of Field Robotics 37 (3) (2020) 362–386. doi:10.1002/rob.21918.

[4] A. Gupta, A. Anpalagan, L. Guan, A. S. Khwaja, Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues, Array 10 (2021) 100057. doi:10.1016/j.array.2021.100057.

[5] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, K. O. Arras, Human motion trajectory prediction: A survey, The International Journal of Robotics Research 39 (8) (2020) 895–935. doi:10.1177/0278364920917446.

[6] S. Mozaffari, O. Y. Al-Jarrah, M. Dianati, P. Jennings, A. Mouzakitis, Deep learning-based vehicle behavior prediction for autonomous driving applications: A review, IEEE Transactions on Intelligent Transportation Systems (2020) 1–15doi:10.1109/TITS.2020.3012034.

[7] A. Miglani, N. Kumar, Deep learning models for traffic flow prediction in autonomous vehicles: A review, solutions, and challenges, Vehicular Communications 20 (2019) 100184. doi:10.1016/j.vehcom.2019.100184.

[8] X. Di, R. Shi, A survey on autonomous vehicle control in the era of mixed-autonomy: From physics-based to AI-guided driving policy learning, Transportation Research Part C: Emerging Technologies 125 (2021) 103008. doi:10.1016/j.trc.2021.103008.

[9] J. Dean, G. S. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M. Ranzato, A. Senior, P. Tucker, et al., Large scale distributed deep networks, in: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, NIPS'12, Curran Associates Inc., Red Hook, NY, USA, 2012, p. 1223–1231.

[10] J. Konečnỳ, H. B. McMahan, D. Ramage, P. Richtárik, Federated optimization: Distributed machine learning for on-device intelligence, arXiv e-prints (2016). arXiv:1610.02527.

[11] S. Savazzi, M. Nicoli, V. Rampa, Federated learning with cooperating devices: A consensus approach for massive IoT networks, IEEE Internet of Things Journal 7 (5) (2020) 4641–4654. doi:10.1109/JIOT.2020.2964162.

[12] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, et al., Advances and open problems in federated learning, arXiv e-prints (2019). arXiv:1912.04977.

[13] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, B. Arcas, Communication-efficient learning of deep networks from decentralized data, arXiv e-prints (2016). arXiv:1602.05629.

[14] S. Savazzi, M. Nicoli, M. Bennis, S. Kianoush, L. Barbieri, Opportunities of federated learning in connected, cooperative, and automated industrial systems, IEEE Communications Magazine 59 (2) (2021) 16–21. doi:10.1109/MCOM.001.2000200.

[15] O. A. Wahab, A. Mourad, H. Otrok, T. Taleb, Federated machine learning: Survey, multi-level classification, desirable criteria and future directions in communication and networking systems, IEEE Communications Surveys Tutorials (2021) 1–1doi:10.1109/COMST.2021.3058573.

[16] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, Y. Gao, A survey on federated learning, Knowledge-Based Systems 216 (2021) 106775. doi:10.1016/j.knosys.2021.106775.

[17] T. Li, A. K. Sahu, A. Talwalkar, V. Smith, Federated learning: Challenges, methods, and future directions, IEEE Signal Processing Magazine 37 (3) (2020) 50–60. doi:10.1109/MSP.2020.2975749.

[18] Q. Xia, W. Ye, Z. Tao, J. Wu, Q. Li, A survey of federated learning for edge computing: Research problems and solutions, High-Confidence Computing (2021) 100008doi:10.1016/j.hcc.2021.100008.

[19] H. T. Nguyen, V. Sehwag, S. Hosseinalipour, C. G. Brinton, M. Chiang, H. Vincent Poor, Fast-convergent federated learning, IEEE Journal on Selected Areas in Communications 39 (1) (2021) 201–218. doi:10.1109/JSAC.2020.3036952.

[20] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečný, S. Mazzocchi, H. B. McMahan, T. V. Overveldt, D. Petrou, D. Ramage, J. Roselander, Towards federated learning at scale: System design, arXiv e-prints (2019). arXiv:1902.01046.

[21] M. Chen, H. V. Poor, W. Saad, S. Cui, Convergence time optimization for federated learning over wireless networks, IEEE Transactions on Wireless Communications 20 (4) (2021) 2457–2471. doi:10.1109/TWC.2020.3042530.

[22] W. Liu, L. Chen, Y. Chen, W. Zhang, Accelerating federated learning via momentum gradient descent, IEEE Transactions on Parallel and Distributed Systems 31 (8) (2020) 1754–1766. doi:10.1109/TPDS.2020.2975189.

[23] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, K. Chan, Adaptive federated learning in resource constrained edge computing systems, IEEE Journal on Selected Areas in Communications 37 (6) (2019) 1205–1221. doi:10.1109/JSAC.2019.2904348.

[24] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, V. Smith, Federated optimization in heterogeneous networks, arXiv e-prints (2020). arXiv:1812.06127.

[25] V. Smith, C.-K. Chiang, M. Sanjabi, A. Talwalkar, Federated multi-task learning, arXiv e-prints (2018). arXiv:1705.10467.

[26] Y. Liu, Y. Kang, C. Xing, T. Chen, Q. Yang, A secure federated transfer learning framework, IEEE Intelligent Systems 35 (4) (2020) 70–82. doi:10.1109/MIS.2020.2988525.

[27] J. Daily, A. Vishnu, C. Siegel, T. Warfel, V. Amatya, GossipGraD: Scalable deep learning using gossip communication based asynchronous gradient descent, arXiv e-prints (2018). arXiv:1803.05880.

[28] H. Xing, O. Simeone, S. Bi, Federated learning over wireless device-to-device networks: Algorithms and convergence analysis, arXiv e-prints (2021). arXiv:2101.12704.

[29] 3GPP TS 22.886 v16.2.0, 3rd Generation Partnership Project; technical specification group services and system aspects; study on enhancement of 3GPP support for 5G V2X services (Release 16) (2018).

[30] P. K. Singh, S. K. Nandi, S. Nandi, A tutorial survey on vehicular communication state of the art, and future research directions, Vehicular Communications 18 (2019) 100164. doi:10.1016/j.vehcom.2019.100164.

[31] S. Samarakoon, M. Bennis, W. Saad, M. Debbah, Distributed federated learning for ultra-reliable low-latency vehicular communications, IEEE Transactions on Communications 68 (2) (2020) 1146–1159. doi:10.1109/TCOMM.2019.2956472.

[32] Y. Qi, M. S. Hossain, J. Nie, X. Li, Privacy-preserving blockchain-based federated learning for traffic flow prediction, Future Generation Computer Systems 117 (2021) 328–337. doi:10.1016/j.future.2020.12.003.

[33] X. Liang, Y. Liu, T. Chen, M. Liu, Q. Yang, Federated transfer reinforcement learning for autonomous driving, arXiv e-prints (2019). arXiv:1910.06001.

[34] D. Ye, R. Yu, M. Pan, Z. Han, Federated learning in vehicular edge computing: A selective model aggregation approach, IEEE Access 8 (2020) 23920–23935. doi:10.1109/ACCESS.2020.2968399.

[35] X. Zhou, W. Liang, J. She, Z. Yan, K. Wang, Two-layer federated learning with heterogeneous model aggregation for 6g supported internet of vehicles, IEEE Transactions on Vehicular Technology (2021) 1–1doi:10.1109/TVT.2021.3077893.

[36] Y. M. Saputra, D. Nguyen, H. T. Dinh, T. X. Vu, E. Dutkiewicz, S. Chatzinotas, Federated learning meets contract theory: Economic-efficiency framework for electric vehicle networks, IEEE Transactions on Mobile Computing (2020) 1–1doi:10.1109/TMC.2020.3045987.

[37] Y. Lu, X. Huang, K. Zhang, S. Maharjan, Y. Zhang, Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles, IEEE Transactions on Vehicular Technology 69 (4) (2020) 4298–4311. doi:10.1109/TVT.2020.2973651.

[38] S. R. Pokhrel, J. Choi, Federated learning with blockchain for autonomous vehicles: Analysis and design challenges, IEEE Transactions on Communications 68 (8) (2020) 4734–4746. doi:10.1109/TCOMM.2020.2990686.

[39] K. Wang, S. P. Xu, C.-M. Chen, S. H. Islam, M. M. Hassan, C. Savaglio, P. Pace, G. Aloi, A trusted consensus scheme for collaborative learning in the edge ai computing domain, IEEE Network 35 (1) (2021) 204–210. doi:10.1109/MNET.011.2000249.

[40] J. Posner, L. Tseng, M. Aloqaily, Y. Jararweh, Federated learning in vehicular networks: Opportunities and solutions, IEEE Network 35 (2) (2021) 152–159. doi:10.1109/MNET.011.2000430.

[41] A. M. Elbir, B. Soner, S. Coleri, Federated learning in vehicular networks, arXiv e-prints (2020). arXiv:2006.01412.

[42] L. Barbieri, S. Savazzi, M. Nicoli, Decentralized federated learning for road user classification in enhanced v2x networks, in: 2021 IEEE International Conference on Communications Workshops (ICC Workshops), 2021, pp. 1–6. doi:10.1109/ICCWorkshops50388.2021.9473581.

[43] R. Q. Charles, H. Su, M. Kaichun, L. J. Guibas, PointNet: Deep learning on point sets for 3D classification and segmentation, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 77–85. doi:10.1109/CVPR.2017.16.

[44] ETSI, Intelligent Transportation Systems (ITS); Vehicular Communications; Basic Set of Applications; Analysis of the Collective Perception Service (CPS); Release 2, Technical Report 103 562, European Telecommunications Standards Institute (ETSI), version 2.1.1 (Dec. 2019).

[45] G. Thandavarayan, M. Sepulcre, J. Gozalvez, Cooperative perception for connected and automated vehicles: Evaluation and impact of congestion control, IEEE Access 8 (2020) 197665–197683. doi:10.1109/ACCESS.2020.3035119.

[46] J. Yosinski, J. Clune, Y. Bengio, H. Lipson, How transferable are features in deep neural networks?, arXiv e-prints (2014). arXiv:1411.1792.

[47] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv e-prints (2014). arXiv:1412.6980.

[48] IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments, IEEE Std 802.11p-2010 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, IEEE Std 802.11n-2009, and IEEE Std 802.11w-2009) (2010) 1–51doi:10.1109/IEEESTD.2010.5514475.

[49] 3GPP TR 22.885 v14.0.0, 3rd Generation Partnership Project; technical specification group services and system aspects; study on LTE support for Vehicle to Everything (V2X) services (Release 14) (2015).

[50] 3GPP TS 22.186 v16.2.0, 3rd Generation Partnership Project; technical specification group services and system aspects; study on enhancement of 3GPP support for 5G V2X services (Release 16) (2019).

[51] C. R. Storck, F. Duarte-Figueiredo, A survey of 5G technology evolution, standards, and infrastructure associated with vehicle-to-everything communications by internet of vehicles, IEEE Access 8 (2020) 117593–117614. doi:10.1109/ACCESS.2020.3004779.

[52] Z. Li, T. Yu, R. Fukatsu, G. K. Tran, K. Sakaguchi, Towards safe automated driving: Design of software-defined dynamic mmwave V2X networks and PoC implementation, IEEE Open Journal of Vehicular Technology 2 (2021) 78–93. doi:10.1109/OJVT.2021.3049783.

[53] G. Noh, J. Kim, S. Choi, N. Lee, H. Chung, I. Kim, Feasibility validation of a 5G-enabled mmwave vehicular communication system on a highway, IEEE Access 9 (2021) 36535–36546. doi:10.1109/ACCESS.2021.3062907.

[54] W. Kim, Experimental demonstration of mmWave vehicle-to-vehicle communications using IEEE 802.11ad, Sensors 19 (9) (2019). doi:10.3390/s19092057.

[55] A. Loch, A. Asadi, G. H. Sim, J. Widmer, M. Hollick, mm-Wave on wheels: Practical 60 GHz vehicular communication without beam training, in: 2017 9th Int. Conf. COMSNETS, 2017, pp. 1–8. doi:10.1109/COMSNETS.2017.7945351.

[56] T. S. Rappaport, S. Sun, R. Mayzus, H. Zhao, Y. Azar, K. Wang, G. N. Wong, J. K. Schulz, M. Samimi, F. Gutierrez, Millimeter wave mobile communications for 5G cellular: It will work!, IEEE Access 1 (2013) 335–349. doi:10.1109/ACCESS.2013.2260813.

[57] B. Bertenyi, 5G evolution: What's next?, IEEE Wireless Communications 28 (1) (2021) 4–8. doi:10.1109/MWC.2021.9363048.

[58] A. Ghosh, A. Maeder, M. Baker, D. Chandramouli, 5G evolution: A view on 5G cellular technology beyond 3GPP Release 15, IEEE Access 7 (2019) 127639–127651. doi:10.1109/ACCESS.2019.2939938.

[59] Z. MacHardy, A. Khan, K. Obana, S. Iwashina, V2X access technologies: Regulation, research, and remaining challenges, IEEE Communications Surveys Tutorials 20 (3) (2018) 1858–1877. doi:10.1109/

COMST.2018.2808444.

[60] D. Tagliaferri, M. Brambilla, M. Nicoli, U. Spagnolini, Sensor-aided beamwidth and power control for next generation vehicular communications, IEEE Access 9 (2021) 56301–56317. doi:10.1109/ACCESS.2021.3071726.

[61] W. Jiang, B. Han, M. A. Habibi, H. D. Schotten, The road towards 6G: A comprehensive survey, IEEE Open Journal of the Communications Society 2 (2021) 334–366. doi:10.1109/OJCOMS.2021.3057679.

[62] Q. Bi, Ten trends in the cellular industry and an outlook on 6G, IEEE Communications Magazine 57 (12) (2019) 31–36. doi:10.1109/MCOM.001.1900315.

[63] Y. Lu, X. Zheng, 6G: A survey on technologies, scenarios, challenges, and the related issues, Journal of Industrial Information Integration 19 (2020) 100158. doi:10.1016/j.jii.2020.100158.

[64] M. Brambilla, M. Nicoli, G. Soatti, F. Deflorio, Augmenting vehicle localization by cooperative sensing of the driving environment: Insight on data association in urban traffic scenarios, IEEE Transactions on Intelligent Transportation Systems 21 (4) (2020) 1646–1663. doi:10.1109/TITS.2019.2941435.

[65] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, O. Beijbom, nuScenes: A multimodal dataset for autonomous driving, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 11618–11628. doi:10.1109/CVPR42600.2020.01164.

[66] C. Bernardini, M. R. Asghar, B. Crispo, Security and privacy in vehicular communications: Challenges and opportunities, Vehicular Communications 10 (2017) 13–28. doi:https://doi.org/10.1016/j.vehcom.2017.10.002.
URL https://www.sciencedirect.com/science/article/pii/S2214209617300803

[67] S. Sharma, B. Kaushik, A survey on internet of vehicles: Applications, security issues & solutions, Vehicular Communications 20 (2019) 100182. doi:https://doi.org/10.1016/j.vehcom.2019.100182.
URL https://www.sciencedirect.com/science/article/pii/S2214209619302293

[68] L. Liu, S. Lu, R. Zhong, B. Wu, Y. Yao, Q. Zhang, W. Shi, Computing systems for autonomous driving: State of the art and challenges, IEEE Internet of Things Journal 8 (8) (2021) 6469–6486. doi:10.1109/JIOT.2020.3043716.