Decentralized Federated Learning for Road User Classification in Enhanced V2X Networks

Luca Barbieri[1],Stefano Savazzi[2],Monica Nicoli[1]
[1]*Politecnico di Milano, Milan, Italy*, [2]*Consiglio Nazionale delle Ricerche, Milan, Italy*

***Abstract*—Federated Learning (FL) techniques are emerging in the automotive context to support connected automated driving services. Yet, when applied to vehicular use cases, conventional centralized FL policies show some drawbacks in terms of latency and scalability. This paper focuses on decentralized FL solutions, which attempt to overcome such limitations, by introducing a distributed computing architecture: vehicles exchange the parameters of a shared Machine Learning (ML) model via V2V links, without the need of a central orchestrator. Sharing all ML parameters, however, might not be feasible when minimal V2X bandwidth usage is required or the model is highly complex (e.g., extremely deep networks) as in advanced scenarios for high levels of automation. We thus propose a modular decentralized FL solution and we discuss its application to road user classification in a cooperative vehicular sensing use case. The proposed FL solution performs the point cloud processing of Lidar sensor inputs using a PointNet compliant architecture. It enables the exchange of a subset of the model parameters, namely selected ML model layers, optimized for communication efficiency, convergence and accuracy. We use real sensor data extracted from a publicly available dataset to validate the method, focusing on non-uniform scenarios where sensor data are highly unbalanced across the connected vehicles. For all cases, FL is shown to outperform the ego-sensing approach with minimal bandwidth usage.**

***Index Terms*—Federated Learning, Connected automated driving, V2X, Artificial Intelligence, Distributed processing**

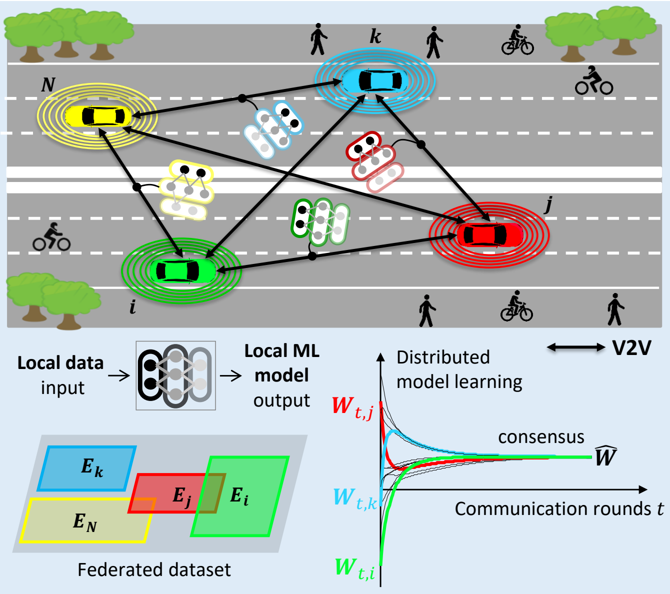


Fig. 1. Decentralized FL architecture: vehicles collectively optimize the ML model by sharing the parameters via V2V communication links.

I. Introduction

Distributed Machine Learning (DML) is a key enabling technology for connected automated driving (CAD) where networked vehicles, with increased level of intelligence and autonomy, are expected to cooperate to improve safety, efficiency and driving comfort. CAD relies on big-data-driven training of large-size Machine Learning (ML) models for several automated functions [1], as well as ultra-reliable low-latency Vehicle to Everything (V2X) interactions with road infrastructure (V2I) and other vehicles (V2V) for cooperative sensing/maneuvering tasks. In such context, the integration of DML techniques [2] with vehicles acting as distributed learners is expected to enable faster, more accurate and flexible training as well as novel decision-making opportunities.

In conventional DML paradigms, the training procedure is supervised by a central orchestrator, e.g., a road side unit (RSU) or a mobile edge cloud (MEC). This is in charge of distributing/collecting raw data to/from the vehicles, learning the ML model from such data, and finally sending back the ML model updates to the participating nodes. In spite of the promising features DML exhibits, privacy concerns may arise as the data is distributed pre-training to all participants, which may be intercepted by performing adversarial attacks. Over the past few years, Federated Learning (FL) [3]–[6], is being actively considered for overcoming the limitations of DML. Rather than exchanging raw data, in FL the interconnected devices share locally trained instances of the ML model parameters, i.e., the weights and biases of the Neural Networks (NN). Furthermore, the data never leaves the holder which makes the overall system more privacy conservative. However, conventional FL techniques require a Parameter Server (PS), namely a central orchestrator to aggregate, update and distribute the local model parameters. With many client connections, PS may slow down the overall training process. Decentralized FL techniques [5], [7] bring further advances as they rely on a fully distributed network architecture rather than the PS. Interconnected devices cooperate with each other by exchanging the model updates with their respective neighbors and implementing a consensus policy.

FL is increasingly being applied to a variety of applications. However, few studies have been reported for automated driving use cases. Connected automated vehicles are expected

This work is supported by the Project BASE5G (Broadband Interfaces and Services for Smart Environments enabled by 5G technologies) under grant POR-FESR 2014-2020, European Regional Development Funding for the Lombardy Region.

to handle huge data streams up to 1 Gbps [8] to support advanced services and improve situational awareness. As the level of automation increases, sharing such large volumes of data become unfeasible due to the extremely stringent requirements envisioned for full self-driving scenarios. FL has the potential to support low-latency and privacy preserving cooperative driving functionalities, as it does not require raw sensor data sharing among vehicles. In [9], 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 [10], while in [11] a federated transfer reinforcement learning approach is developed for real-time knowledge extraction. Other studies focus on model selection for aggregation [12] or contract-based FL to maximize energy efficiency in electric vehicle networks [13]. Finally, [14] presents an initial feasibility study on FL for vehicular networks.

In this paper, we propose the use of a decentralized FL policy for classification of road user/object (here referred to as road actor) in connected vehicular scenarios. The proposed FL tool implements the ML model parameter exchange over V2V links and is shown to augment the recognition capability compared with ego-vehicle approaches. Lidar sensor's readings are used as input to a Deep Learning (DL) model, namely PointNet [15], which is in charge of inferring the actual type of road actors. Considering FL over deep architectures, typically adopted for solving automated vehicle tasks, a novel, modular approach is proposed, in which the FL process can be enabled across a variable number of ML model layers. 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, in practical settings characterized by highly imbalanced data distributions across vehicles, or non-independent and identically distributed (non-IID) information. 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. Compared to raw sensor's data sharing, the proposed approach heavily reduces the amount of information to be exchanged and thereby the communication latency, so as to comply with the requirements of next-generation V2X communications for high levels of automation [8].

The paper is organized as follows: Sec. II introduces the proposed FL algorithm, Sec. III explains in details the use case considered throughout the paper, Sec. IV describes the setup used for the validation and Sec. V presents the numerical results. Finally, Sec. VI draws conclusions.

II. Decentralized Federated Learning

In decentralized FL, federated vehicles combine local models with neighboring ones by distributed averaging. Next, they update (adapt) the combined models using an assigned optimizer. The FL process runs for a number of communication rounds and ends when a consensus is obtained, namely when local models converge to a common representation [7].

Considering a deep NN composed of N layers, the goal of FL is to learn a global model $\mathbf{W} = \mathbf{W}^{(N)}$ that encapsulates the parameters of the NN, from the available input data $\mathbf{x}$ and for all N layers. The input data samples collected by the vehicles consist here of Lidar sensor readings. The overall dataset $\mathcal{E}$ is unevenly distributed across K vehicles $\{\mathcal{E}_k\}_{k=1}^{K}$, in addition, the number of examples E_k held by each vehicle is much lower compared to the overall pool of examples, i.e., $E_k \ll E = |\mathcal{E}|$. Samples can be generally expressed in the form (x_h, y_h) with $h = 1, \ldots, E_k$, where y_h is the desired output, namely information about true road actors, corresponding to the input x_h. In FL the parameters $\mathbf{W}$ can be learned by applying a minimization procedure to any finite-sum objective function $L(\mathbf{W})$ as:

$$\min_{\mathbf{W}} L(\mathbf{W}) = \min_{\mathbf{W}} \underbrace{\sum_{i=1}^{K} \rho_k \times L_k(\mathbf{W})}_{L(\mathbf{W})}, \tag{1}$$

where $\rho_k = E_k/E$ and $L_k(\mathbf{W})$ is the local loss of device k:

$$L_k(\mathbf{W}) = \frac{1}{E_k} \sum_{h=1}^{E_k} \ell(x_h, y_h; \mathbf{W}) \tag{2}$$

and $\ell(x_h, y_h; \mathbf{W})$ is the loss computed over the examples (x_h, y_h) when the parameters $\mathbf{W}$ hold. For the specific case of road actor classification, described in Sect. IV, $\ell(x_h; y_h; \mathbf{W})$ is the loss presented in [15], computed with C classes, corresponding here to road actor categories.

As depicted in Fig. 1, in decentralized FL the vehicles learn the global objective (1) by relying only on local computations and parameters $\mathbf{W}$ exchange with neighbors through V2V communications. To limit the model size and thus the communication overhead, we propose a modular approach in which the federated optimization is implemented over a subset of model layers $Q \leq N$, while the remaining $N - Q$ ones are learned using local data only and the chosen optimizer. In particular, for an assigned number of layers Q, the federated optimization targets a collaborative learning of the Q layers closest to the outputs, as more sensitive to unbalanced distributions since they learn features specific to the dataset considered [16]. For Q layers, the corresponding model parameters $\mathbf{W} = \mathbf{W}^{(Q)}$ to be exchanged on each communication round are now

$$\mathbf{W}^{(Q)} = [w_{N-Q}^{\mathrm{T}}, b_{N-Q}, \cdots, w_N^{\mathrm{T}}, b_N], \tag{3}$$

with w_n and b_n being the NN weights and biases of the layer $n = N - Q, \cdots, N$. The optimized number Q of layers for the chosen study is analyzed in Sect. V.

The vehicles are assumed as connected according to a directed graph $\mathcal{G} = (\mathcal{V}, \xi)$, where $\mathcal{V} = 1 \ldots, K$ are the set of nodes and ξ are the edges. We also denote the set of neighbors to k as $\mathcal{N}_{\overline{k}} = \mathcal{N}_k \setminus \{k\}$. This set may change over time to reflect the current connectivity graph, according to the relative distance among vehicles and to account for intermittent connections. For vehicle k and time t, using the same notations in (3), let the local model parameters subject to decentralized FL process be represented as $\mathbf{W}_{t,k} = \mathbf{W}_{t,k}^{(Q)}$, on every communication round $t > 0$ the vehicle k fuses the parameters received from its neighbors $i \in \mathcal{N}_{\overline{k}}$ as:

$$\psi_{t,k} = \sigma_{k,k}\mathbf{W}_{t,k} + \sum_{i\in\mathcal{N}_{\overline{k}}} \sigma_{i,k}\mathbf{W}_{t,i}, \tag{4}$$

where $\sigma_{i,k}$ are the mixing weights for the received models which are chosen as $\sigma_{i,k} = \frac{E_i}{\sum_{i\in\mathcal{N}_{\overline{k}}} E_i}$ [7]. Once the aggregation phase is completed, the vehicle k adapts its fused $\psi_{t,k}$ parameters to minimize the local loss, namely $L_k(\mathbf{W}_{t,k})$ of (2). In this paper, we consider the Adam optimizer [17], being the preferred option for PointNet based architectures [15]. Therefore, using the fused parameters $\psi_{t,k}$, the last stage is implemented as $\mathbf{W}_{t+1,k} = \psi_{t,k} - \Delta\psi_{t,k}$ with:

$$\begin{cases} \Delta\psi_{t,k} = \mu_t \cdot \dfrac{\sqrt{1-\beta_2^t}}{1-\beta_1^t} \cdot \dfrac{\mathbf{m}_{t+1,k}}{\sqrt{\mathbf{v}_{t+1,k}} + \delta} \\ \mathbf{m}_{t+1,k} = \beta_1\mathbf{m}_{t,k} + (1-\beta_1)\nabla L_{t,k}(\psi_{t,k}) \\ \mathbf{v}_{t+1,k} = \beta_2\mathbf{v}_{t,k} + (1-\beta_2)\nabla^2 L_{t,k}(\psi_{t,k}) \end{cases}, \tag{5}$$

where $\mathbf{m}_{t+1,k}$ and $\mathbf{v}_{t+1,k}$ are the estimates of the first and second moment of the gradients $\nabla L_{t,k}(\psi_{t,k})$ at round $t+1$. Other parameters $\beta_1, \beta_2 \in [0,1)$, μ_t and δ are defined in [17]. Model adaptation in (5) is computed over a mini-batch B of local training examples. The new parameters $\mathbf{W}_{t+1,k}$ are forwarded to the the neighbors of vehicle k and a new round starts. This procedure is iterated until the parameters $\mathbf{W}_{t,k}$ converge to the desired loss values.

III. Vehicular Scenario and Dataset

In this section, we present the considered scenario and the dataset used for the experimental validation of the proposed FL approach. We assume that V vehicles participate in the FL process and are able to collaboratively optimize the ML model for road actors classification by exchanging the NN parameters through V2X links. They do so by applying the decentralized FL algorithm presented in Sec. II. As depicted in the Fig. 2, the road actor classification system in each vehicle uses the Lidar point clouds as inputs for object detection and it consists of two subsystems. First, the input raw Lidar data are processed independently on each vehicle to extract a bounding box that provides object segmentation information. We thus aggregate 10 Lidar sweeps and extract the point clouds that fall within the boxes. Next, on a second stage (classification sub-system), these segmented point clouds are used as inputs for classification using the NN PoinNet model. On each FL round, federated optimization is applied to the classification sub-system while bounding box processing is implemented locally without applying any model federation. This choice is shown as practical enough to minimize the communication overhead.

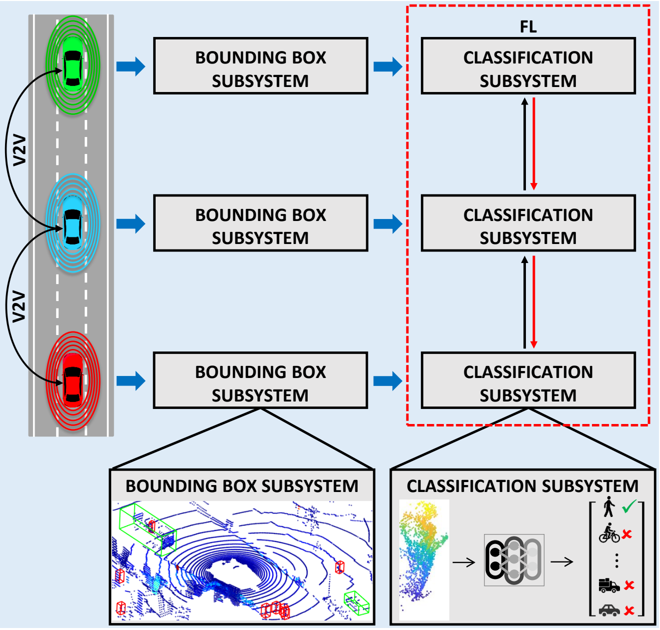
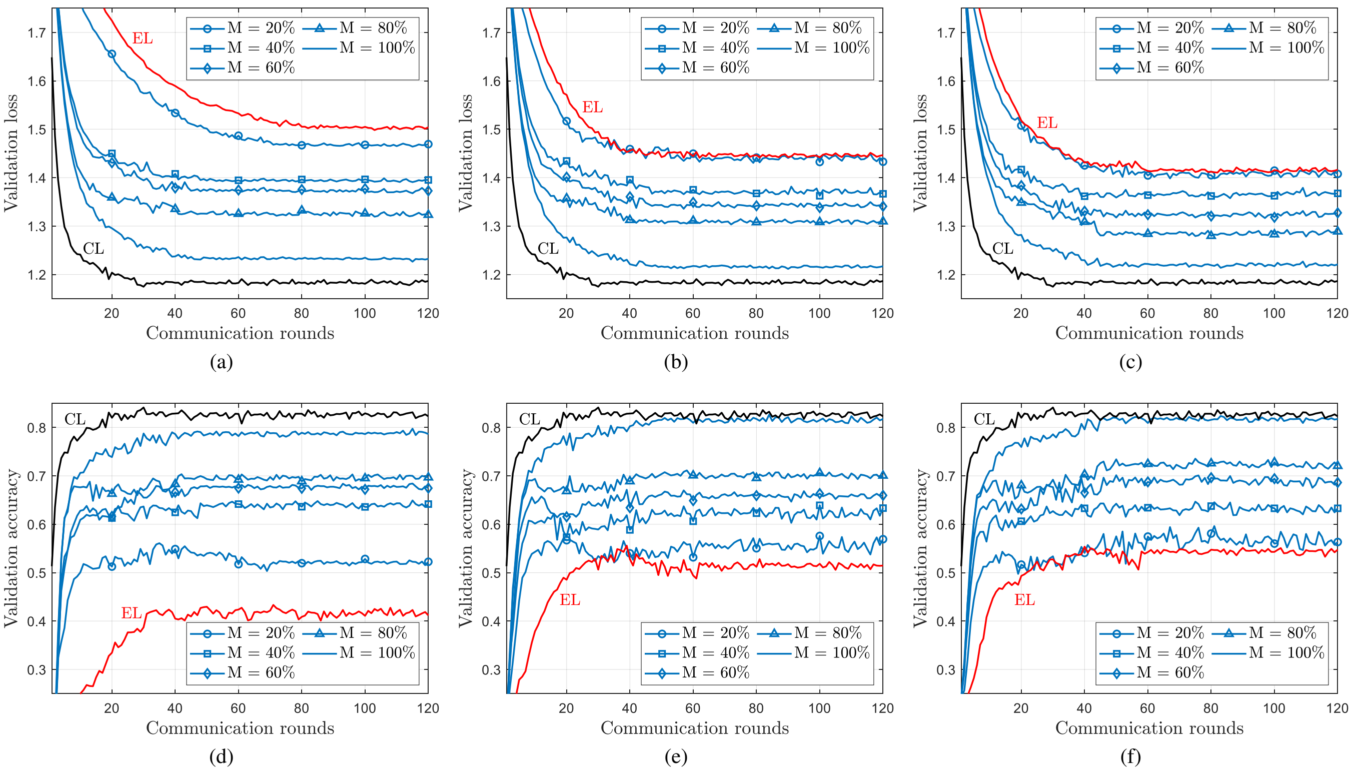


Fig. 2. Road actor classificator consisting of 2 subsystems: i) the bounding box segmentation and ii) the classifier based on the PointNet model. FL is applied for distributed training of the PointNet NN model parameters.

The dataset considered for the experimental validation is nuScenes [18]. It 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 2Hz to ensure high accuracy. Each annotated object consists of a 3D bounding box as well as the object's category. Overall, 23 road actors categories are available, ranging from large vehicles, i.e., trailer and truck, to little objects, i.e., traffic cones.

IV. Processing and Modeling of Sensors Data for Road Actor Classification

FL have been simulated using the Lidar dataset of nuScenes. A virtual environment allows to deploy vehicles acting as virtual mobile devices that learn over a configurable subset of data. The overall database used for classification is composed by 9000 examples in total, equally divided for each one of the considered categories: pedestrian, car, bus, bicycle, barrier and traffic cones. The h-th training example contains the point cloud x_h that fall into the box and the corresponding category. To simulate FL performance the database can be distributed to the virtual devices according to an arbitrary policy (IID and non-IID). Before feeding the training examples to the local NN PoinNet model, 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.



(a) (b) (c)

(d) (e) (f)

Fig. 3. Analysis of the proposed FL approach for unbalanced IID data partitioning. Results are presented considering the validation loss for: (a) ρ_0 = 2%, (b) ρ_0 = 4%, (c) ρ_0 = 6%. Similarly, the validation accuracy is reported for: (d) ρ_0 = 2%, (e) ρ_0 = 4%, (f) ρ_0 = 6%.

The PointNet ML model [15] is used for 3D shape classification and segmentation from point cloud data. 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 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 now considers only $C = 6$ output neurons. Furthermore, the Batch Normalization (BN) layers apply a momentum factor of 0.9. In the proposed implementation, FL is 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 $N = 20$ while the total number of model parameters that can be exchanged is 40855.

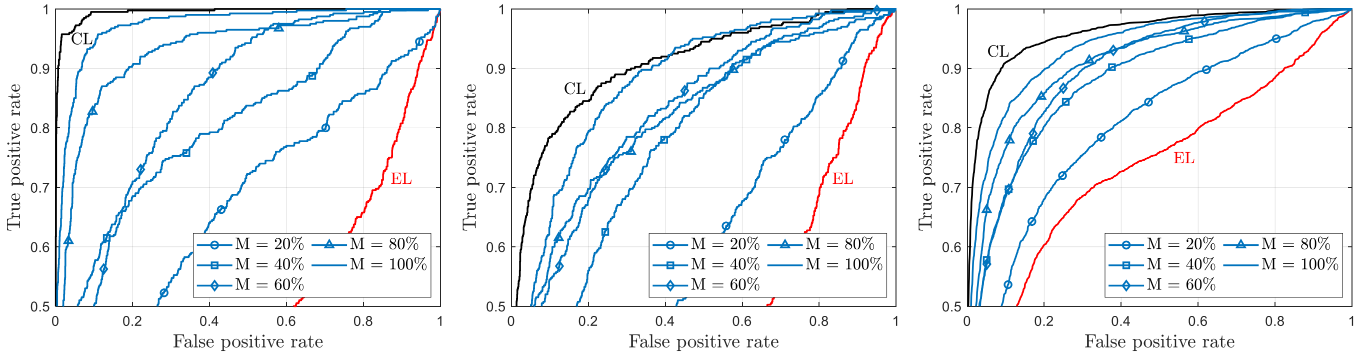
V. Numerical Results

This section presents the numerical results and the validation of the proposed FL tools. Subsec. V-A presents the results for the IID case, while Subsec. V-B focuses on the non-IID one. Finally, Subsec. V-C discusses the communication aspects of the FL approach.

A. IID Results

The validation of the proposed FL approach is performed over a virtual environment composed by $V = 4$ vehicles, connected according to a random network topology. In particular, at every communication round, each vehicle chooses randomly two other neighbors from the set $\mathcal{N}_{\overline{k}}$ and sends its parameters for model averaging. Before the training starts, the overall database is partitioned and distributed among the vehicles according to a predefined policy, either being IID or non-IID. Local optimization is performed over mini-batches of $B = 30$ examples each. The Adam optimizer of (2) is configured as $\mu_t = 5{\cdot}10^{-5}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\delta = 1{\cdot}10^{-7}$. The main metrics used for performance evaluation are the validation loss, accuracy and Receiver Operating Characteristic (ROC) curves, computed over a separated dataset consisting of 2400 examples, equally split among the classes. As introduced before, decentralized FL is implemented by considering a variable number $Q < N$ of layers as input for distributed optimization. In particular, in what follows performances are analyzed by varying the fraction $M = Q/N$ of the layers subject to federation. This ranges from $M = 20\%$, (i.e., corresponding to the last $Q = 4$ layers at the output of the NN model) up to $M = 100\%$. FL is also compared against opportunistic behaviours, referred to as Ego Learning (EL), where vehicles do not participate in FL and rely solely on their local data for minimizing (2). To further analyze the proposed technique, we also compare it with a DML implementation, referred to as Centralized Learning (CL), where the training process is coordinated by a single entity, namely a data center.

The first analysis focuses on how IID, but unbalanced, data partitioning affects the FL performances. In this scenario, the



(a)

(b)

(c)

Fig. 4. Analysis of the proposed FL approach for non-IID data partitioning. ROC curves are presented for: (a) Class 1 (barrier), (b) Class 2 (traffic cone) and (c) average over all classes.

local dataset contains the same number of examples for each class but the overall percentage varies considering different vehicles. More specifically, one vehicle, i.e. $k = 0$, retains a small percentage $\rho_0 = E_0/E < 25\%$ of the database, while the remaining ones hold $\rho_k = 25\%, \forall k > 0$. In the following, performances are presented for the vehicle ($k = 0$) retaining the lowest amount of data as more critical.

Fig. 3 presents the logarithmic validation loss (top) and validation accuracy (bottom) for $\rho_0 = 2\%$ (Fig. 3a and Fig. 3d), $\rho_0 = 4\%$ (Fig. 3b and Fig. 3e) and $\rho_0 = 6\%$ (Fig. 3c and Fig. 3f). FL is able to outperform EL in the majority of the cases considered, while approaching the performances of CL when $M = 100\%$. For $\rho_0 = 2\%$, it is sufficient to exchange only the $M = 20\%$ of the layers to increase the classification accuracy by almost 10% compared to EL. Considering now $\rho_0 = 4\%$ and $\rho_0 = 6\%$, the mutual exchange of the $M = 20\%$ of the NN layers provides a small increase in accuracy with respect to EL. However, by increasing the number of shared layers to $M = 40\%$, the accuracy improvements exceed 10%.

Focusing on the performances of FL with varying M, it is evident that the classification accuracy increases with the number of layers exchanged among interconnected vehicles. This is reasonable since the vehicle retaining the lowest amount of data can adapt its NN model parameters cooperating with vehicles that have a refined set of features as they hold a much larger pool of examples. As the number of shared parameters grows, the vehicle holding the smallest dataset can fine tune its model to better adhere to the overall features learned by other vehicles, and consequently increase the classification accuracy. Clearly, all layers should be exchanged to maximize accuracy, however, limited communication resources in the V2X network may not permit to do so and a trade-off between bandwidth usage and classification accuracy should be carefully analyzed. This further aspect is better investigated in Sect. V-C.

B. Non-IID Results

The second scenario considered focuses on how non-IID data affects the performances of the proposed FL approach. In this context, the vehicle $k = 0$ stores $\rho_0 = 16.7\%$ of the overall database, equally divided among 4 classes, excluding Class 1 (barrier) and Class 2 (traffic cone), while the others retain $\rho_k = 25\%, \forall k > 0$, evenly distributed among all available classes. Performances are again presented for the vehicle holding the non-IID dataset, i.e., $k = 0$.

TABLE I
COMPARISON OF AUC VALUES

	EL	M = 20%	M = 40%	M = 60%	M = 80%	M = 100%	CL
Class 1	0.41	0.65	0.80	0.84	0.93	0.97	0.99
Class 2	0.41	0.55	0.76	0.82	0.82	0.88	0.91
Average	0.73	0.80	0.88	0.90	0.91	0.94	0.96

Fig. 4 highlights the performances of the proposed FL approach ranging from $M = 20\%$ up to $M = 100\%$, and using now the ROC as metric for comparison with CL and EL. To show the benefits of employing FL, the ROC curves are computed for Class 1 (Fig. 4a), Class 2 (Fig. 4b) and averaged over all classes (Fig. 4c). Table I reports the Area Under The Curve (AUC) values for all methods, focusing on Class 1, Class 2 and the average among classes. Results indicate that FL is able to achieve a substantially higher true positive rate compared to EL, still approaching the performances of CL when all layers are exchanged during federated optimization. For Class 1, a strong performance improvement with respect to EL can be obtained when $M = 20\%$, or equivalently sharing only $Q = 4$ layers: the AUC is 0.65 compared to 0.41 of EL. Class 2 requires more layers to be exchanged as the performances of FL and EL are similar when $M \leq 20\%$. Considering average performances, EL can be outperformed by a large margin when $M \geq 40\%$. Good classification scores, i.e., when AUC $>= 0.80$, can be achieved with $M = 60\%$. This heavily reduces the communication resources compared to $M = 100\%$ at the cost of a small loss of accuracy.

C. Communication Requirements Considerations

The proposed modular FL approach allows to optimize the number of layers to be exchanged during the distributed training procedure. The communication overhead and resources used thus heavily depend on this choice. To show the benefits of FL, here we evaluate the payload required for transmitting the model parameters and compare it with CL that requires the fusion of raw data on the data center. Considering the overall database, each example is comprised of 2048 3D point clouds, and CL requires a payload of 221.18 Mbytes assuming single precision encoding (32 bits per sample) as typically used for Lidar point clouds. On the other hand, FL with M ranging from $M = 20\%$ to $M = 100\%$ needs 12710 up to 40855 model parameters to be exchanged, respectively, with corresponding payloads of 101.68 Kbytes and 326.84 Kbytes. Encoding of each parameter has double precision (64 bits per sample). Notice that compression of model parameters is not implemented, therefore further improvements are possible. In all cases, FL substantially reduces the communication requirements compared to CL, while providing a comparable accuracy. Same considerations apply to non-IID data partitioning case as the ROC curves, and consequently the AUC values are comparable. This suggests that FL approaches could heavily reduce the communication overhead. Finally, it should be also noted that the payload for FL does not depend on the data size, as opposed to CL, making it the preferred choice for vehicular applications handling big data structures.

Focusing now on the required spectral efficiency or data rate, the total transmission time needed for exchanging the model parameters should be comparable with the time required for running the Adam local optimizer. This prevents the communication rounds to delay the overall FL process. For the considered study, an average computational time of 6 s is observed for local optimization. Therefore, the corresponding minimum required communication data rate ranges from 135.57 Kbit/s, for $M = 20\%$, up to 435.79 Kbit/s when $M = 100\%$. Sending raw Lidar data in 6 s would require a massive 295.33 Mbit/s for distributing the overall database. In case of information sharing for high/full automated driving [8], up to 35 Mbit/sec can be allocated for exchanging Lidar point cloud data. Considering the proposed use case of Sec. III, and the required payload of 221.18 Mbytes, CL will thus require several rounds for transmitting all the dataset, even using the full available bandwidth, and introduce large delays during the training process. On the other hand, the proposed approach keeps the bandwidth usage to a minimum, allowing the V2X network to allocate resources where they are most needed.

VI. Conclusions

In this paper, we developed a decentralized FL technique for road actor classification in vehicular networks. Real Lidar point clouds extracted from the nuScenes dataset are processed by a PointNet based architecture for classification. A modular approach has been proposed to optimize the number of exchanged layers to increase the classification accuracy while minimizing the bandwidth usage. Different data partitioning strategies, i.e, unbalanced IID and non-IID, have been used to benchmark the performances of the proposed approach, and compare against opportunistic (EL) behaviours and centralized (CL) implementations. In all cases, FL has been shown to outperform EL while obtaining comparable accuracy with respect to CL when all parameters are exchanged. Moreover, FL requires minimal communication resources and complies with the stringent requirements of the extended sensing use cases of next-generation V2X services for high levels of automaton. Further research activities will target the employment of quantization/sparisification operations applied on the model parameters to improve communication efficiency and/or the inclusion of gradient exchange mechanisms to further promote convergence.

References

[1] S. Grigorescu *et al.*, "A survey of deep learning techniques for autonomous driving," *Journal of Field Robotics*, vol. 37, no. 3, pp. 362–386, 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21918

[2] J. Dean *et al.*, "Large scale distributed deep networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'12. Red Hook, NY, USA: Curran Associates Inc., 2012, p. 1223–1231.

[3] J. Konečný *et al.*, "Federated Optimization: Distributed Machine Learning for On-Device Intelligence," *arXiv e-prints*, Oct. 2016. [Online]. Available: https://arxiv.org/abs/1610.02527

[4] S. Savazzi *et al.*, "Opportunities of Federated Learning in Connected, Cooperative, and Autonomous Industrial Systems," *IEEE Communications Magazine*, vol. 52, no. 2, February 2021. [Online]. Available: https://arxiv.org/abs/2101.03367

[5] P. Kairouz *et al.*, "Advances and Open Problems in Federated Learning," *arXiv e-prints*, Dec. 2019. [Online]. Available: https://arxiv.org/abs/1912.04977

[6] H. Brendan McMahan *et al.*, "Communication-Efficient Learning of Deep Networks from Decentralized Data," *arXiv e-prints*, Feb. 2016. [Online]. Available: https://arxiv.org/abs/1602.05629

[7] S. Savazzi *et al.*, "Federated Learning With Cooperating Devices: A Consensus Approach for Massive IoT Networks," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4641–4654, 2020.

[8] 3GPP, "Study on enhancement of 3GPP Support for 5G V2X Services (Release 16) ," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 22.886, 12 2018, version 16.2.0.

[9] S. Samarakoon *et al.*, "Distributed federated learning for ultra-reliable low-latency vehicular communications," *IEEE Transactions on Communications*, vol. 68, no. 2, pp. 1146–1159, 2020.

[10] Y. Liu *et al.*, "Privacy-preserving traffic flow prediction: A federated learning approach," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7751–7763, 2020.

[11] X. Liang *et al.*, "Federated Transfer Reinforcement Learning for Autonomous Driving," *arXiv e-prints*, Oct. 2019. [Online]. Available: https://arxiv.org/abs/1910.06001

[12] D. Ye *et al.*, "Federated learning in vehicular edge computing: A selective model aggregation approach," *IEEE Access*, vol. 8, pp. 23 920–23 935, 2020.

[13] Y. M. Saputra *et al.*, "Federated learning meets contract theory: Economic-efficiency framework for electric vehicle networks," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2020.

[14] A. M. Elbir *et al.*, "Federated Learning in Vehicular Networks," *arXiv e-prints*, Jun. 2020. [Online]. Available: https://arxiv.org/abs/2006.01412

[15] R. Q. Charles *et al.*, "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.

[16] J. Yosinski *et al.*, "How transferable are features in deep neural networks?" *arXiv e-prints*, Nov. 2014. [Online]. Available: https://arxiv.org/abs/1411.1792

[17] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv e-prints*, Dec. 2014. [Online]. Available: https://arxiv.org/abs/1412.6980

[18] H. Caesar *et al.*, "nuscenes: A multimodal dataset for autonomous driving," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 618–11 628.